

Desenvolupament d'un *framework* de validació per aplicacions web amb dependències d'interacció humana amb caixers automàtics

Treball Final de Grau - Memòria

Autor: Claudi Cervelló Nogués
Director: Ignasi González Gómez
Ponent: Dolors Costal Costa (Dep. ESSI)
Titulació: Grau en Enginyeria Informàtica
Data: 28 de juny del 2016
Especialitat: Enginyeria del Software
Centre: Facultat d'Informàtica de Barcelona(FIB)
Universitat Politècnica de Barcelona (UPC) - Barcelona Tech

Vull agrair a tots els qui, en algun moment o altre, han estat implicats en aquest projecte, el seu esforç, la seva paciència i la seva dedicació.

En primer lloc al director i a la ponent pels consells i orientacions al llarg del projecte.

En segon lloc als membres de l'equip TSB_ATM_FULL_TEAM per l'ajuda i la bona feina.

I per últim, i molt especialment, al meu entorn més proper pel suport incondicional i la confiança.

A tots, moltes gràcies!

Abstract – Català

L'*E2E-framework* que presentem en aquesta Treball de final de grau és un *framework* per donar resposta al repte de realitzar *testing* funcional de la futura aplicació que s'està desenvolupant per als caixers automàtics de l'entitat financera britànica TSB.

Aquest *testing* s'ha de poder dur a terme *end-to-end*, assegurant l'execució de tot el sistema però sense disposar físicament d'un caixer automàtic. Per tant, cal simular la interacció que realitzaria un usuari final amb el nou *software* del caixer. L'objectiu és, d'una banda, poder comprovar que el funcionament del sistema és correcte i, a més a més, poder fer aquestes comprovacions de forma automàtica, fàcilment repetible i executant-les en un breu termini de temps.

Per dur a terme la construcció de l'*E2E-framework* ha calgut analitzar i entendre el context, la necessitat i els requisits; elaborar una planificació temporal del projecte i un pressupost; també s'han identificat els riscos, els *stakeholders*, l'impacte social i el marc legal.

La solució desenvolupada cobreix tot l'abast del problema plantejat i ha permès assolir els objectius proposats aportant un *framework* per poder validar aplicacions web amb dependències d'interacció humana amb caixers automàtics.

Abstract - Castellano

El *E2E-framework* que presentamos en este Trabajo de final de grado es un *framework* para dar respuesta al reto de realizar un *testing* funcional de la nueva aplicación que se está desarrollando para los cajeros automáticos de la entidad financiera británica TSB.

Este *testing* debe llevarse a cabo *end-to-end*, asegurando la ejecución de todo el sistema pero sin disponer físicamente de un cajero automático. Por lo tanto, es necesario simular la interacción que realiza un usuario final con el nuevo *software* del cajero. El objetivo es, por una parte, poder comprobar que el funcionamiento del sistema es correcto y, por otra, hacer estas comprobaciones de forma automática, fácilmente repetible y ejecutándolas en un breve período de tiempo.

Para realizar la construcción del *E2E-framework* ha sido necesario analizar y entender el contexto, la necesidad y los requisitos; elaborar una planificación temporal del proyecto y un presupuesto; también se han identificado los riesgos, los *stakeholders*, el impacto social y el marco legal.

La solución desarrollada cubre todo el ámbito del problema planteado y alcanza los objetivos planteados, todo ello mediante un *framework* que permite validar aplicaciones web con dependencias de interacción humana en cajeros automáticos.

Abstract - English

The *E2E-framework* presented in this Degree's final thesis is a framework to respond to the challenge of performing functional testing of the future application which is being developed for British financial institution TSB's ATMs.

This testing must be carried out end-to-end, ensuring the execution of the entire system without physically having an ATM. Therefore, it is necessary to simulate the interaction between an end user and the new ATM's software. The objective, on the one hand is to check that the system is operating correctly and, on the other hand, to be able to make these checks automatically, easily repeatable and running them in a short period of time.

In E2E-framework construction had been necessary to analyse and understand the context, the needs and the requirements; to develop a project plan and budget; also identify risks, stakeholders, social impact and legal framework.

The solution developed covers the full scope of the problem achieving the objectives and providing a framework which allows us to validate web applications with human interaction with ATMs.

Índex de continguts

1. Introducció.....	9
2. Context	11
2.1. Introducció a l'arquitectura d' <i>ATM for TSB</i>	11
2.1.1. Diagrama del context	15
3. Formulació del problema	17
4. Stakeholders	18
5. Estat de l'art.....	19
6. Abast	22
6.1. Riscos i obstacles.....	24
7. Metodologia de treball	26
7.1. Valoració d'alternatives i pla d'acció	27
8. Planificació.....	28
8.1. Descripció de les tasques.....	28
8.2. Diagrama de Gantt.....	30
9. Especificació de requisits.....	33
9.1. Requisits funcionals	33
9.1.1. Diagrama de casos d'ús	33
9.1.2. Descripció dels casos d'ús	38
9.1.3. Diagrama conceptual.....	49
9.2. Requisits no funcionals	53
9.2.1. Requisits de <i>look and feel</i>	53
9.2.2. Requisits d'usabilitat i humanitat.....	53
9.2.3. Requisits de rendiment	54
9.2.4. Requisits operacionals i ambientals	54
9.2.5. Requisits de manteniment i suport	54
9.2.6. Requisits de seguretat	55
9.2.7. Requisits culturals.....	56
10. Arquitectura de l' <i>E2E-framework</i>	57
10.1. Emulador.....	57
10.1.1. Arquitectura del SDL.....	58
10.1.2. Arquitectura de l'emulador	60
10.1.3. Funcionament de l'emulador	61
10.2. Mòdul de respostes	63
10.2.1. Arquitectura del mòdul de respostes.....	63
10.2.2. Funcionament del mòdul de respostes	63
10.3. Arquitectura de tests	64
10.3.1. Escriptura i automatització dels tests.....	64
10.3.2. Tipus de codificació de les expressions regulars	66
11. Proves	69
11.1. Proves de l'emulador i del mòdul de càrrega	69
11.2. Proves <i>end-to-end</i>	69
12. Tecnologies i eines emprades	72

13. Gestió econòmica	74
13.1. Identificació i estimació dels costos.....	74
13.1.1. Recursos humans.....	74
13.1.2. Recursos hardware	74
13.1.3. Recursos software	75
13.1.4. Despeses generals	75
13.1.5. Imprevistos	75
13.1.6. Contingència.....	76
13.1.7. Cost total	76
13.2. Seguiment i control del projecte	76
14. Sostenibilitat i compromís social.....	77
14.1. Sostenibilitat econòmica.....	77
14.2. Sostenibilitat social	78
14.3. Sostenibilitat ambiental.....	78
15. Aspectes legals.....	79
16. Assoliment de les competències tècniques	80
17. Conclusions i treball futur.....	81
Glossari	83
Bibliografia.....	85

Índex d'il·lustracions

Il·lustració 1: Part anterior d'un caixer.....	11
Il·lustració 2: Part posterior d'un caixer	12
Il·lustració 3: Components lògics caixer automàtic.....	13
Il·lustració 4: Diagrama del context.....	15
Il·lustració 5: Captures de pantalla del Global JAM Simulator durant un test manual..	20
Il·lustració 6: Abast de la solució (software caixer).....	22
Il·lustració 7: Abast de la solució (infraestructura)	23
Il·lustració 8: Esquema d'esdeveniments Scrum propis.....	27
Il·lustració 9: Diagrama de Gantt part I (de Novembre a Març)	31
Il·lustració 10: Diagrama de Gantt part II (d'Abril a Juny)	32
Il·lustració 11: Diagrama de casos d'ús de l'Emulador	34
Il·lustració 12: Diagrama de casos d'ús de l'API Service.....	35
Il·lustració 13: Diagrama de casos d'ús dels Test	36
Il·lustració 14: Diagrama de casos d'ús del E3E-framework	37
Il·lustració 15: Diagrama conceptual.....	52
Il·lustració 16: Diagrama de les comunicacions entre components del E2E-framework	57
Il·lustració 17: Diagrama UML de la capa Websocket i la interfície de la capa Actions. 58	
Il·lustració 18: Diagrama UML de la capa Actions i les interfícies de la capa del Core..	59
Il·lustració 19: Diagrama UML de la capa Core i les interfícies de la capa de Integration	59
Il·lustració 20: Diagrama UML de la capa Integration i les classes de la capa de JSI	60
Il·lustració 21: Diagrama UML de la capa Integration i les classes de l'emulador	61

Il·lustració 22: Diagrama del funcionament de l'emulador.....	61
Il·lustració 23: Diagrama mòdul de respostes.....	63
Il·lustració 24: Diagrama de l'arquitectura dels tests.....	64
Il·lustració 25: Exemple d'una feature	65
Il·lustració 26: Exemple de codificació d'expressions regulars	66
Il·lustració 27: Menú.....	67
Il·lustració 28: Informe post execució test end-to-end: expressions regulars.....	70
Il·lustració 29: Informe post execució test end-to-end taula de tests	70
Il·lustració 30: Informe post execució test end-to-end detall d'un test	71
Il·lustració 31: Tecnologies emprades per component	73

Índex de taules

Taula 1: Comparació possibles solucions	21
Taula 2: Comparació possibles solucions II	21
Taula 3: Riscos	25
Taula 4: Tecnologies i eines emprades I (desenvolupament)	72
Taula 5: Tecnologies i eines emprades II (gestió de codi)	72
Taula 6: Tecnologies i eines emprades III (gestió documentació i equip).....	72
Taula 7: Tecnologies i eines emprades IV (realització de la memòria)	73
Taula 8: Despeses recursos humans.....	74
Taula 9: Despeses recursos hardware	74
Taula 10: Despeses recursos software	75
Taula 11: Despeses generals.....	75
Taula 12: Cost total.....	76

1. Introducció

El document que teniu a les mans és un Projecte de Final de Grau d'Enginyeria Informàtica, de l'especialitat d'Enginyeria del Software i modalitat B (Conveni CCE[1]), a la Facultat d'Informàtica de Barcelona a la Universitat Politècnica de Catalunya.

Com s'ha dit aquest projecte pertany a la modalitat B, és a dir, el projecte s'està realitzant en el marc d'un conveni de cooperació educativa per a la realització de pràctiques acadèmiques externes en entitats col·laboradores. Aquest conveni es va iniciar el 28 setembre del 2015 amb l'empresa consultora Everis[2], tot i que no seria fins el 10 de novembre del 2015 quan va començar el projecte.

Aquest projecte forma part d'un altre projecte de més abast, el qual pertany al sector de la banca i que s'anomena *ATM for TSB* i consisteix en el desenvolupament i implantació d'una infraestructura per a caixers automàtics. És rellevant mencionar el projecte *ATM for TSB* perquè el projecte descrit en aquest document se centra en els components desenvolupats per l'equip d'*ATM for TSB*.

En concret, l'objectiu del projecte consisteix en desenvolupar un *framework*¹ anomenat *E2E-framework* que primàriament permeti dur a terme un *testing* del nou software destinat als caixers automàtics de TSB que es desenvolupa en el marc d'*ATM for TSB*.

Aquest *testing* s'ha de poder dur a terme *end-to-end*, assegurant l'execució de tot el sistema però sense disposar físicament d'un caixer automàtic. Per tant, cal simular la interacció que realitzaria un usuari final amb el nou software del caixer. L'objectiu és, d'una banda, poder comprovar que el funcionament del sistema és correcte i, a més a més, poder fer aquestes comprovacions de forma automàtica, fàcilment repetible i executant-les en un breu termini de temps.

Per dur a terme la construcció de l'*E2E-framework* ha calgut analitzar i entendre el context, la necessitat i els requisits; elaborar una planificació temporal del projecte i un pressupost; també s'han identificat els riscos, els *stakeholders*, l'impacte social i el marc legal.

¹ *Framework* defineix, en termes generals, un conjunt de conceptes, pràctiques i criteris per solucionar un problema en concret i aquestes serveixen per encarar i resoldre problemes d'índole similar.

A continuació s'introdueix el context del projecte, rellevant els punts de més interès i situant al lector en el projecte d'*ATM for TSB*.

En primer lloc, detallarem el problema que es vol resoldre, i analitzaré els *stakeholders* i estat del art, per poder proporcionar una visió del que es vol aconseguir i quins són els objectius.

Seguidament marcarem els límits projecte: en primer lloc parlant abast del projecte i la seva metodologia de treball. I en segon lloc situant-lo en el temps mitjançant la planificació.

Després d'aquests punt passarem a la part més tècnica del document introduint els requisits funcionals i l'arquitectura de la solució. Per acabar aquesta part exposarem les proves i per últim les tecnologies emprades en el projecte.

Encararem la recta final amb la gestió econòmica presentant el pressupost seguit de l'estudi de sostenibilitat i el compromís social. Segueixen els aspectes legals que emmarquen el projecte. Finalment s'esmenten les competències tècniques treballades en aquest projecte amb el propòsit d'autoavaluar si realment s'han assolit i treballat, i acabarem amb les conclusions i la visió de futur.

2. Context

Com a resultat de l'adquisició de l'entitat bancària britànica TSB per part del Banc Sabadell, Everis va rebre l'encàrrec per millorar la infraestructura de negoci del banc TSB i adaptar aquesta a la infraestructura que actualment està en funcionament al Banc Sabadell. D'aquesta manera el projecte d'*ATM for TSB* és la part corresponent a la millora i adaptació de la infraestructura dels caixers automàtics que l'entitat bancària TSB té al Regne Unit.

Tot seguit s'explicarà la feina que l'equip d'*ATM for TSB* està realitzant per poder entendre millor problema que intenta resoldre el projecte que descriu aquest document. Primerament, una introducció als components lògics d'un caixer automàtic, de manera que, al explicar tant el problema com la solució, el lector estigui familiaritzat amb els diferents components.

2.1. Introducció a l'arquitectura d'*ATM for TSB*

En l'actualitat un caixer automàtic es compon d'un conjunt de *hardware*, similar al d'un portàtil, el qual executa un *software* que orquestra tots els dispositius segons les lectures que li proporcionen aquests dispositius del estat dels seus sensors. L'objectiu final és completar de forma segura una transacció interactuant amb un usuari humà.

A la il·lustració 1 la imatge de la dreta es mostra la part anterior, la que veuria el client, trobem els elements més comuns: els botons de selecció, la pantalla, el dispensador de bitllets, la impressora de rebuts, el lector de targetes i el teclat segur (on s'introdueix el codi PIN).



Il·lustració 1: Part anterior d'un caixer

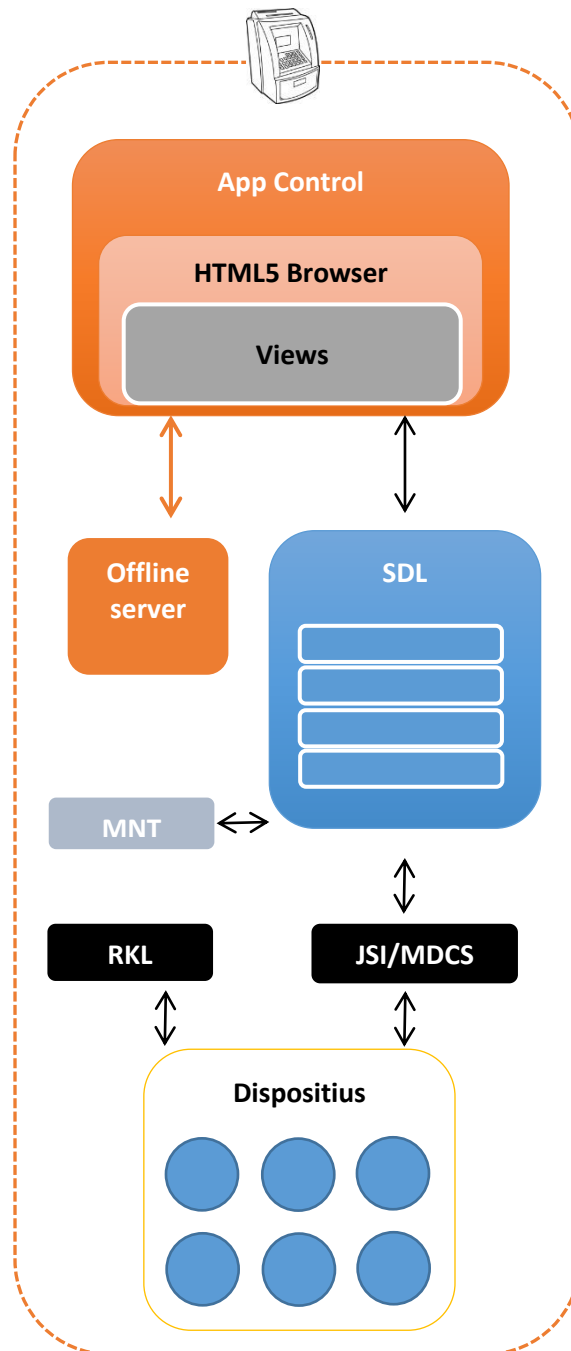
Pel que fa a la part posterior del caixer, il·lustració 2, trobem a la part inferior la zona de la caixa forta la qual conte tots els contenidors que emmagatzemen els diferents tipus de bitllets i a la part superior hi ha un segona pantalla dedicada a tasques de manteniment i monitorització a més a més del hardware i els dispositius.



Il·lustració 2: Part posterior d'un caixer

Per plasmar tots aquest components mencionats la següent il·lustració mostra un esquema dels components lògics que acabem d'esmentar. Aquest esquema ens aporta una visió general dels components que resideixen en un caixer automàtic tan hardware, mencionant anteriorment, com software.

S'introdueixen i es defineix la seva funcionalitat de tots aquest components ja que més endavant es farà referencia a aquests components.



Il·lustració 3: Components lògics caixer automàtic

A continuació es descriu breument la funcionalitat dels components de l'esquema anterior:

- **App Control:** s'encarrega de gestionar que tots els components estiguin actius durant una transacció i realitza peticions al SDL. A més a més encapsula el navegador (mostrat al esquema com "HTML5 browser") que és el que finalment mostra les vistes al client.

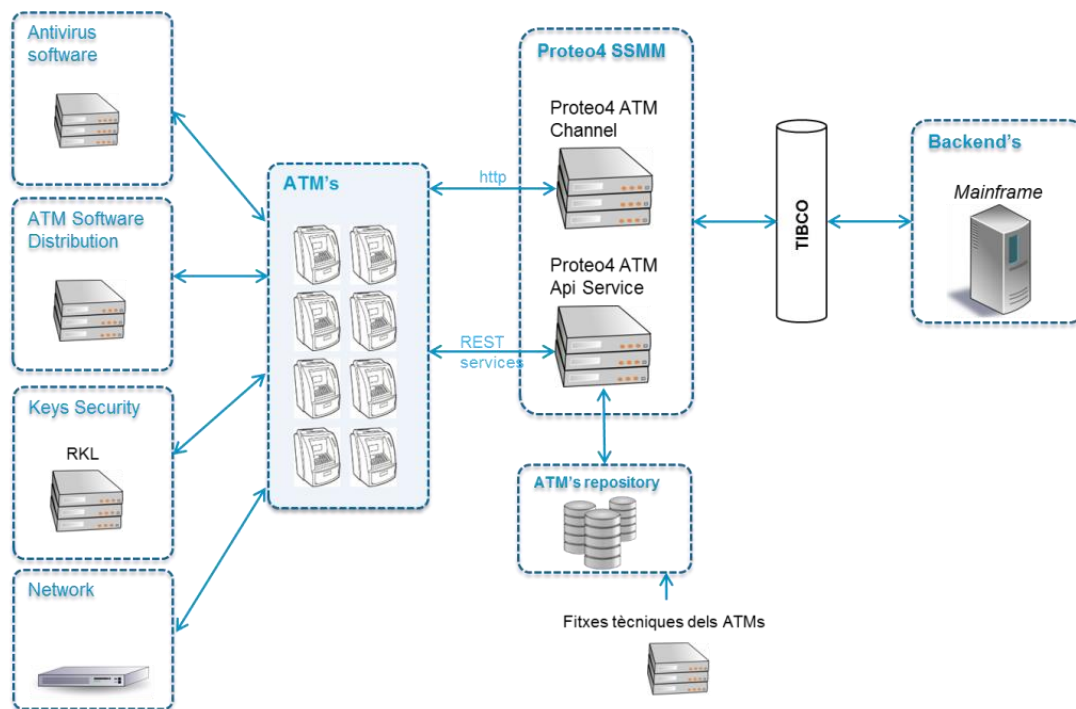
- SDL (Serveis Dispositius Locals): aquest component és l'encarregat d'encapsular les peticions que fa el client, provinents de l'App Control, sobre dispositius hardware del caixer.
- Offline server: es tracta del servidor utilitzat quan es perd la connexió o bé quant es produeix un error durant una transacció. És, per tant, el servidor que conté les vistes del flux d'error i la lògica de com gestionar l'error (targetes, bitllets, ...)
- JSI i MDCS: són els components encarregats d'integrar el SDL amb els dispositius físics.

Dispositius: són tots aquells dispositius que mitjançant la informació que recullen els seus sensors permeten a un usuari completar una transacció: els botons de selecció, el dispensador de bitllets, la impressora de rebuts, el lector de targetes i el teclat segur on s'introdueix el codi PIN, etc.

- RKL o Remote Key Loader: agent de carregar de claus d'encryptació de forma remota.
- MNT o Maintenance: aplicació per realitzar tasques de manteniment del caixer normalment utilitzada per un tècnic o operador de l'entitat.

2.1.1. Diagrama del context

Una vegada disposem d'una visió global dels components que resideixen un caixer automàtic, a partir de la il·lustració 3, passarem a descriure i analitzar la infraestructura necessària més enllà de la pròpia del caixer: això ajudarà a situar-nos per veure l'abast de la solució proposada i a més a més poder veure l'abast de la solució que està realitzant l'equip d'*ATM for TSB*.



Il·lustració 4: Diagrama del context

- ATM: ATM o Automated Teller Machine és caixer automàtic de l'entitat o en el cas del diagrama el conjunt de caixers automàtics de l'entitat.
- Antivirus software: servidor del antivirus que els caixers tenen instal·lat en local.
- ATM software distribution: mecanisme per distribuir *software* als caixers remotament.
- Keys Security (RKL): servidor que proporciona claus d'encryptació a caixers.

- Proteo 4 SSM (o Sistemas Medis): és la part de la infraestructura dedicada a proporcionar serveis als caixers que no tenen o no poden tenir en local
 - Proteo4 ATM Channel: API que gestiona l'aplicació que visualitza el client.
 - Proteo4 ATM API Service: API que gestiona les peticions dels caixers conta TIBCO.
- TIBCO: canal per fer peticions i rebre respostes d'informació sensible, com per exemple el número de compte o de targeta, que té l'entitat financera.
- Backend: Base de dades de l'entitat bancària.

3. Formulació del problema

En qualsevol desenvolupament de software, la part de *testing* és imprescindible: aquesta part contribueix, entre moltes altres aspectes, a garantir la qualitat del software, però sobre tot garantir que a nivell funcional no hi han errors. Bona part d'aquests errors surten a la llum amb els test unitaris (metodologia on conjunt de tests garanteixen la funcionalitat a nivell local d'una petita part del codi, normalment parts d'un mateix component).

La major dificultat apareix quan cal fer *testing* funcional. El *testing* funcional és una metodologia on un conjunt de test o proves funcionals garanteixen una funcionalitat completa del sistema. Quan ens referim a que "garanteixen una funcionalitat" ens referim a funcionalitats que en última instància són requeriments funcionals, és a dir allò que s'espera que l'usuari final pugui fer

En definitiva, el repte del *testing* funcional, en projectes com el d'*ATM for TSB*, resideix en el fet que el sistema -un caixer automàtic- té unes funcionalitats molt ben definides que cal complir escrupolosament, però la complexitat ve donada pel fet de la interacció humana amb dispositius físics (lector de targetes, el teclat, ...): davant la imprevisibilitat de les accions i fins i tot dels errors que pugui dur a terme l'usuari, cal garantir que el sistema preservi la coherència i consistència de cadascun dels passos, adverteixi i corregeixi els errors adients i eviti accions no desitjades.

Pel grau de complexitat que suposa el testing funcional, no és viable fer totes les proves funcionals manualment, i per tant cal disposar d'un procediment que automatitzi aquesta tasca.

Per tant, en última instància el que es vol aconseguir és poder executar proves funcionals de forma automàtica simulant la interacció d'un humà amb els dispositius del caixer automàtic i el comportament d'aquest sense fer ús d'un caixer automàtic real.

4. Stakeholders

La identificació dels stakeholders és important per l'èxit del projecte. Cal establir comunicació amb tots ells per assegurar que tots ells tenen les expectatives vers el projecte i comprenen amb claredat quines actuacions es duen a terme en cada moment per complir aquestes expectatives o rectificar el més aviat possible en cas de no estar avançant en la direcció d'assolir alguna de les expectatives.

Tot seguit descrivim els stakeholders identificats i els seus requeriments i expectatives envers el projecte.

El promotor del projecte és el gerent del projecte d'*ATM for TSB*. El seu requeriment és disposar d'un mecanisme de proves de regressió automatitzat que permeti identificar errors el més aviat possible. A més a més ha de poder demostrar al client que la solució ha estat testejada.

Els desenvolupadors d'*ATM for TSB: el software de testing* els ajudarà a provar que els canvis i addicions de funcionalitats que facin a l'aplicació funcionen com s'espera i no tenen afectes colaterals. A més a més la màquina d'integració continua podrà passar els test funcionals de forma automàtica detectant possibles problemes.

Els actors no desenvolupadors, és a dir, tots aquells actors no familiaritzats amb el codi com pot ser el client o tot els experts funcionals. Els permetrà veure, gràcies al llenguatge natural dels tests, tot el que s'ha provat i proposar noves proves.

Per últim la pròpia empresa Everis és un actor interessat en aquest projecte de cara a projectes similars, ja que la majoria de components podran ser reutilitzats.

5. Estat de l'art

Per tal de poder disposar del software de testing funcional s'han identificat dos productes comercials candidats a solucionar el problema:

- MDCS una solució desenvolupada per Fujitsu[3]
- JSI una solució desenvolupada per Dynasty Technology Group una empresa del grup Wincor[4]

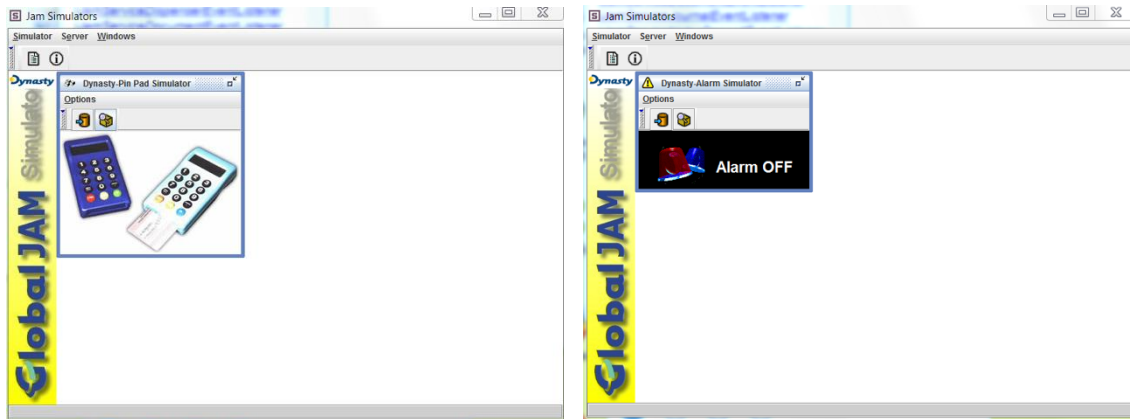
No obstant cap dels dos soluciona el problema en la seva totalitat. Tot seguit introdueixo i comparo aquest dos productes amb els objectius identificats a l'abast.

MDCS va quedar descartat des del principi com a candidat a solucionar el problema plantejat, ja que no es pot accedir al codi perquè és propietat de Fujitsu. Per tant, aquesta solució ens hauria col·locat en una situació de total dependència dels propietaris del software. A més a més seguíem necessitant alguna forma d'emulador que proporcionés dades a MDCS per poder fer proves. En definitiva, no era viable intentar construir un emulador per sota d'una capa sobre la qual no disposem de suficient informació. A més a més quedàvem limitats pel producte, el qual no podem modificar, motiu pel qual el funcionament del nostre software vindria estrictament condicionat per aquest producte.

Per un altre costat, la solució de Wincor(JSI) presenta dos avantatges clars respecte l'anterior. Primer perquè ofereix una API que és programada en el marc del projecte; per tant tenim una part del codi tot i que també quedem condicionats pel que ofereix el proveïdor i no podem realitzar-hi modificacions. En segon lloc disposem d'un emulador dels dispositius hardware, anomenat el Global JAM Simulator[5] imprescindible per solucionar el problema plantejat, malgrat que també és propietat del seu fabricant, Dynasty. Ara bé, no disposem el codi de l'emulador ni sabem com està desenvolupat, motiu pel qual no podem afegir ni modificar el funcionament.

Tot i saber que no disposem del cent per cent del codi aquests productes podrien ser considerats els candidats per solucionar el problema, però no va ser així pels motius següents:

- I. Els test amb tots dos producte són manuals: això significa que, per a cada prova que es vol realitzar, el desenvolupador ha d'interactuar directament amb l'emulador, com es pot observar a la Il·lustració 5.



Il·lustració 5: Captures de pantalla del Global JAM Simulator durant un test manual

- II. No és possible escriure test: per tant no hi ha comprovacions, excepte les que el mateix desenvolupador fa al mateix temps que realitza la prova.
- III. No és possible automatitzar ni l'execució de tests (perquè són manuals), ni les comprovacions de les proves que es pugin fer.

La solució adoptada per fer front al requeriment del testing funcional automatitzat amb traces, ha estat desenvolupar una solució a mida, anomenat *E2E-framework*.

En la següent taula és una comparativa de les solucions considerades anteriorment enfront de la solució a mida.

		Codi			
	Propietat	Disponibilitat del codi font	Possibilitat de modificació	Funcionalitat API	Emulador
MDCS	Fujitsu	no	no	Restringida pel proveïdor	no
JSI	Wincor	no	no	Restringida pel proveïdor	sí
E2E-framework	Everis	sí	sí	Definida per nosaltres	sí

Taula 1: Comparació possibles solucions

	Test			
	Es possible fer tests	Test manuals	Escriure Test	Automatització de Tests
MDCS	no	no	no	no
JSI	sí	sí	no	no
E2E-framework	sí	sí	sí	sí

Taula 2: Comparació possibles solucions II

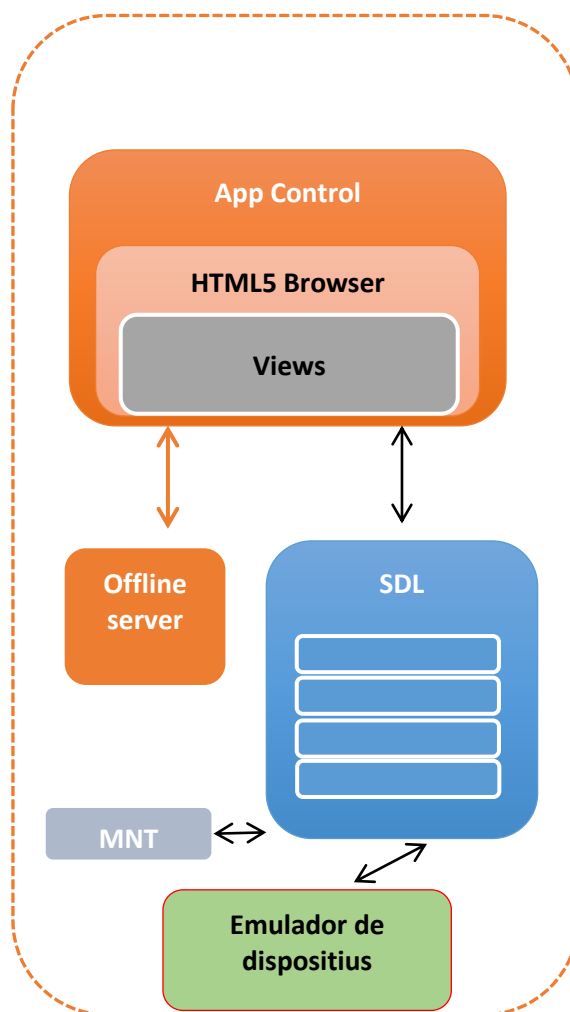
En conclusió, com s'observa en les taules anteriors la naturalesa d'aquest problema porta a una solució a mida, no tan sols pels impediments mencionats anteriorment, sinó que a més a més cal incorporar aquesta solució a l'arquitectura d'*ATM for TSB*, que és una arquitectura a mida.

6. Abast

El repte el qual ens enfrontem és com donar suport a la realització de *testing* funcional en l'aplicació desenvolupada per l'equip d'*ATM for TSB*. Aquest testing s'ha de poder efectuar assegurant l'execució de tot el sistema (o *end-to-end*) sense disposar d'un caixer automàtic, simulant la interacció que realitzaria un usuari final. L'objectiu és poder comprovar que el funcionament del sistema és correcte i per últim poder fer aquestes comprovacions automàticament.

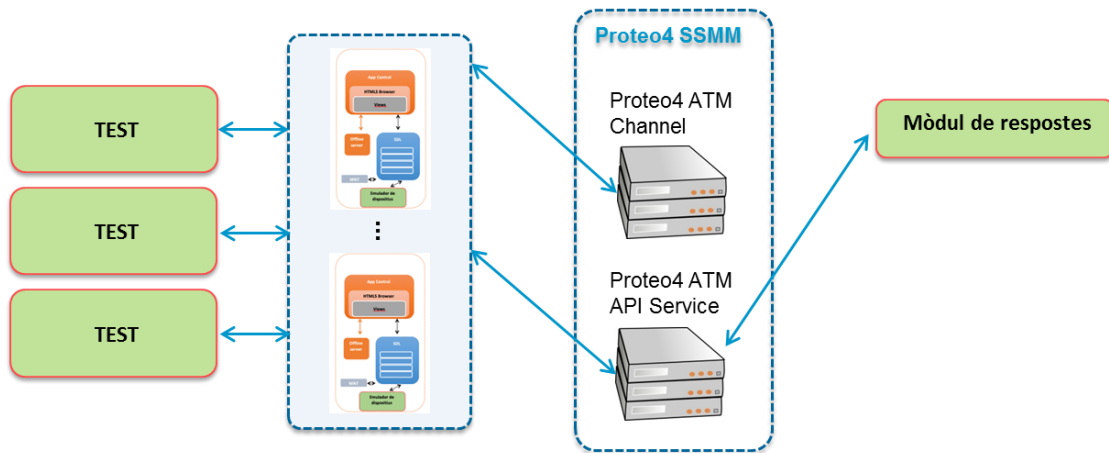
El següent esquema representa de forma molt conceptual els reptes al qual aquest projecte ha de fer front.

Els components de color verd són els proposats per solucionar el problema, en gris es representen els components de l'arquitectura d'*ATM for TSB*. I els entorns indiquen les funcionalitats i requeriments perquè la solució sigui possible.



Il·lustració 6: Abast de la solució (software caixer)

En la il·lustració 7 podem observar l'abast complet de la solució.



Il·lustració 7: Abast de la solució (infraestructura)

Un cop determinat l'abast complet de la solució, representat a la il·lustració 6 i 7, s'han detectat sis subproblemes diferents. A continuació s'identifiquen els sis subproblemes i els seus objectius:

- Creació d'un emulador : analitzar i seguidament desenvolupar un component que imiti el comportament dels dispositius hardware d'un caixer automàtic. A més a més del manteniment d'aquest software, solucionar error i afegir funcionalitats si és necessari. En conclusió anàlisi, desenvolupament, *testing* i manteniment de l'emulador.
- Generació dels patrons d'emulació: aconseguir que l'emulador generi dades el més semblants possibles a les que genera un dispositiu físic utilitzat per un usuari humà. Per fer aconseguir-ho cal incorporar components que interactuin amb l'emulador, els quals acabaran de donar la lògica d'un dispositiu hardware a les dades que emeti l'emulador.
- Creació de les respostes de *backend*: consisteix en disposar de les respostes que enviaria el *backend* en una transacció carregada en el mòdul de resposta del API Services. Aquest component és el que ens permetrà simular les accions i comprovacions que faria l'entitat bancària sobre una transacció.
- Entorn d'execució controlat : integrar l'emulador amb l'arquitectura d'ATM *for TSB*. Aconseguir executar el SDL d'ATM *for TSB* i que aquest consumeixi les dades generades per l'emulador. Un cop assegurat això, el següent pas

és executar tota l'arquitectura d'*ATM for TSB* amb les dades que genera l'emulador.

- Entorn de proves on poder executar i validar tests funcionals : es compon de tres objectius. El primer componer l'entorn de proves, és a dir, instal·lar i preparar totes les llibreries i *frameworks* necessàries per poder llançar un test. El segon és executar tests funcionals els quals realitzen comprovacions. El tercer i últim punt es desprèn del segon: definir totes les expressions regulars per escriure els tests i les seves comprovacions; amb altres paraules: escriure els tests i el que volem comprovar en cada cas.
- Automatitzar l'execució de test funcionals (inclou automatitzar l'entorn d'execució i l'entorn de proves): novament aquest problema es compon de tres objectius, el primer automatitzar l'entorn d'execució i proves via tasques i scripts. El segon assegurar que tots els entorns i components funcionen en sintonia, assegurar el correcte funcionament i en cas contrari solucionar els errors. I per últim proporcionar a l'equip d'*ATM for TSB* la documentació necessària per poder executar test funcionals.

A partir d'aquest punt del document en endavant la solució plantejada, que anomenem *E2E-framework*, es compon de:

- 1) el emulador
- 2) el mòdul de respostes
- 3) els tests
- 4) els patrons d'emulació
- 5) els components que ha desenvolupat l'equip *ATM for TSB*: SDL, App Control, MNT, Offline server i Proteo4 SSMM.

6.1. Riscos i obstacles

Aquest projecte segueix una metodologia de desenvolupament àgil (s'explica extensament aquesta metodologia en el següent apartat) segons la qual els canvis són introduïts en l'abast del projecte. Ara bé, per evitar entrar en desviacions molt extremes la mesura més freqüent de contingència és reforçar l'equip o simplement no estar en disposició d'introduir el canvi en l'abast del projecte.

Els principals riscos i obstacles identificats en aquest projecte es recullen tot seguit / a la taula següent:

Risc identificat	Mesura preventiva	Mesura reactiva
Cancel·lació del projecte	Complir els contracte que es te amb el client i no perdre la seva confiança.	És un risc de cara al projecte en global però no de cara a aquest projecte de final de carrera. Es dissenyaria una prova de concepte per validar els objectius plantejats en aquest document tot hi que el projecte en si no seguís endavant.
Canvis de prioritat o objectius del projecte	Identificar els aspectes que més interessen al client i que aporten més valor a la seva forma de negoci.	Focalitzar l'esforç de l'equip en la nova prioritat. Això pot implicar aturar momentàniament aquest projecte de final de carrera, tot i que un cop alineada la prioritat del client i el projecte es continuaria amb el que va quedar aturat.
Canvis inesperats en l'arquitectura	Programar seguint les recomanacions i estàndards. Evitar lligar parts innecessàries a components d'arquitectura.	Identificar el punt on falla el codi i fer les modificacions necessàries per adaptar el codi a la nova arquitectura perquè torni a funcionar.
Complexitat i forta corba d'aprenentatge de l'arquitectura	Documentar el projecte de manera que tothom que hi participa pugui consultar-la	Formar i donar suport a l'equip per part d'experts en l'arquitectura i per part d'experts funcionals de caixers automàtics.
Mala estimació temporal	Divisió de la feina en tasques més específiques i concises. I estimació temporal d'aquestes tasques per part de tot l'equip.	Avaluar si la desviació és justificada i si cal continuar o parar i reavaluar el que s'està fent. I donat el cas que sigui una situació crítica es destinaran més membres de l'equip a aquesta tasca.

Taula 3: Riscos

7. Metodologia de treball

En el desenvolupament de tot el projecte *ATM for TSB*, se segueix una metodologia de desenvolupament *Scrum*. *Scrum* és una metodologia de desenvolupament àgil i iterativa que permet fer front a problemes complexos maximitzant el valor del producte.

Scrum distingeix tres aspectes: equip, esdeveniments i artefactes.

Pel que fa a l'equip, *Scrum* proposa:

- Product owner: una sola persona encarregada d'introduir i organitzar tasques al *product backlog* (explicat a l'apartat d'artefactes).
- Development team: l'equip de desenvolupament.
- Scrum master: és la persona que té per l'objectiu de maximitzar el valor del producte, assegurant el compliment per part de tot l'equip de la metodologia.

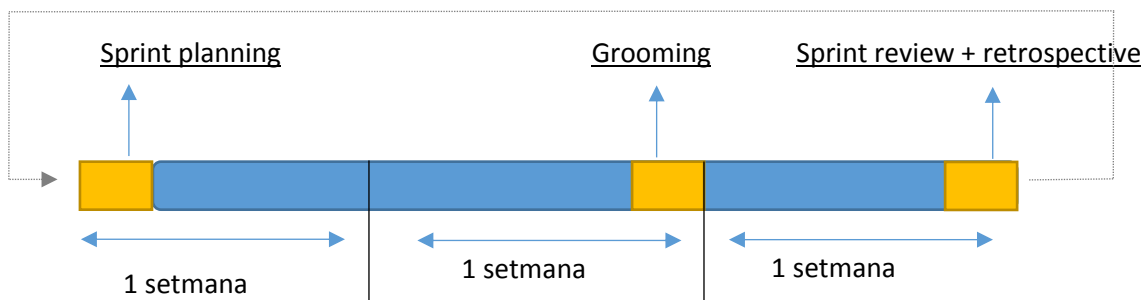
Pel que fa a esdeveniments, *Scrum* proposa:

- Sprint: un període de temps d'un mes o menys on s'incrementa el valor del producte.
- Sprint planning: una reunió abans d'un *sprint* per definir el *sprint backlog* (explicat a l'apartat d'artefactes). En definitiva: planificar la feina a realitzar durant l'*sprint*.
- Daily scrum: una reunió diària de 15 minuts per sincronitzar tot l'equip i crear un pla per les pròximes 24 hores.
- Sprint review: una reunió al final del *sprint* per valorar l'increment de valor del producte i modificar el *product backlog* si es detecten canvis.
- Sprint retrospective: una reunió abans del pròxim *sprint* Planning d'autocrítica de tots els membres, on es proposen millores de cara al següent *sprint*.

Pel que fa a artefactes, *Scrum* proposa:

- Product backlog: Una llista ordenada per prioritat amb totes les tasques a fer per completar el producte.
- Sprint backlog: És el subconjunt de tasques del *product backlog* a realitzar en un *sprint* cada *sprint planning* es torna a fer.

Tot i que existeixen guies oficials de *Scrum*[6] en les que expliquen tots els elements de la metodologia, cada entorn de treball requereix adaptar-los a les seves necessitats. Per tant el projecte segueix aquesta metodologia pel que fa a l'equip i artefactes, ara bé, en relació als esdeveniments s'han introduït certes particularitats: la primera és que un *sprint* són tres setmanes i la segona es que introduïm un esdeveniment més anomenat *Grooming*, al final de la segona setmana, per detectar tasques de cara al pròxim *sprint planning* que han sorgit arrel de les que s'estan realitzant en aquest *sprint*. I per últim la reunió de *sprint review* i *sprint retrospective* les agrupem en una sola al final de cada *sprint*.



Il·lustració 8: Esquema d'esdeveniments Scrum propis

A més a més al cap de 9 setmanes hi ha una release, això vol dir, que s'entrega codi amb valor (amb alguna funcionalitat) al client.

7.1. Valoració d'alternatives i pla d'acció

El projecte segueix un metodologia àgil com l'explicada anteriorment, per tant els possibles problemes seran valorats en les reunions de seguiment (*sprint plan*, *sprint close* i *grooming*) i es prendran les mesures necessàries per poder absorbir el canvi amb un mínim impacte.

L'estimació de tasques té un marge d'hores afegit, no obstant ens permet avançar amb la tasca següent si es finalitza abans del període que havíem estimat.

8. Planificació

Com s'explica a l'apartat anterior, es segueix una metodologia de desenvolupament àgil iterativa. Això implica que cada iteració es planifica en la reunió de *sprint planning* en funció tant del que s'ha fet en la iteració anterior com de quina part prioritza o aporta més valor al client. Les iteracions són de tres setmanes i està previst seguir així fins el gener del 2017.

No obstant això, en el marc d'un projecte de final de grau hi ha una fita o *milestone* de finalitzar com a mínim els components que formen l'emulador (explicat a l'abast del projecte) i tota la documentació de la memòria abans de la defensa del projecte de final de grau.

8.1. Descripció de les tasques

Per assolir el *milestone* esmentat, s'han definit sis macro tasques:

- 1) Familiarització
- 2) Emulador
- 3) Entorn d'execució
- 4) Entorn de proves
- 5) Automatització
- 6) Documentació

Tot seguit es mostra cada una de les macro tasques i les tasques que la conformen.

Macro tasca 1: Familiarització

-Inclou totes aquelles tasques de familiarització amb l'arquitectura que es farà servir i l'entorn de desenvolupament.

Tasques :

- 1.1. Familiarització amb l'arquitectura AMT for TSB
- 1.2. Instal·lació de l'entorn de desenvolupament SDL
- 1.3. Instal·lació de l'entorn de desenvolupament AppChannel

Macro tasca 2: Emulador

-Totes les tasques relacionades amb el desenvolupament del Emulador, tan el propi *core* com components secundaris (ContextLoaderManager).

Tasques :

- 2.1.1. Emulador: Desenvolupar l'emulador per al dispositiu Magstripe
- 2.1.2. Emulador: Test unitaris de l'emulador per al dispositiu Magstripe
- 2.1.3. Emulador: Desenvolupar l'emulador per al dispositiu Receipt
- 2.1.4. Emulador: Test unitaris de l'emulador per al dispositiu Receipt

- 2.1.5. Emulador: Desenvolupar l'emulador per al dispositiu SecureKey
- 2.1.6. Emulador: Test unitaris de l'emulador per al dispositiu Securekey
- 2.1.7. Emulador: Desenvolupar l'emulador per al dispositiu Dispenser
- 2.1.8. Emulador: Test unitaris de l'emulador per al dispositiu Dispenser
- 2.1.9. Emulador: Desenvolupar l'emulador per al dispositiu Device
- 2.1.10. Emulador: Test unitaris de l'emulador per al dispositiu Device
- 2.1.11. Emulador: Desenvolupar l'emulador per al dispositiu MNT
- 2.1.12. Emulador: Test unitaris de l'emulador per al dispositiu MNT
- 2.1.13. Emulador: Desenvolupar l'emulador per al dispositiu RKL
- 2.1.14. Emulador: Test unitaris de l'emulador per al dispositiu RKL
- 2.2.1. ContextLoaderManager: Desenvolupar EndPoint
- 2.2.2. ContextLoaderManager: Desenvolupar Core (carrega de contextos)
- 2.2.3. ContextLoaderManager: Desenvolupar Descodificadors

Macro tasca 3: Entorn d'execució

-Totes les tasques relacionades amb les fitxers d'especificació del comportament de cada una de les unitats del emulador.

Tasques :

- 3.1.1. Contexts: Creació dels contextos per al dispositiu Magstripe
- 3.1.2. Contexts: Creació dels contextos per al dispositiu Receipt
- 3.1.3. Contexts: Creació dels contextos per al dispositiu SecureKey
- 3.1.4. Contexts: Creació dels contextos per al dispositiu Dispenser
- 3.1.5. Contexts: Creació dels contextos per al dispositiu Device
- 3.1.6. Contexts: Creació dels contextos per al dispositiu MNT
- 3.1.7. Contexts: Creació dels contextos per al dispositiu RKL

Macro tasca 4: Entorn de proves

-Totes les tasques relacionades amb la definició i gestió dels components de l'entorn de proves.

Tasques :

- 4.1.1. TestsFuncionals: Definició expressions regulars per al dispositiu Magstripe
- 4.1.2. TestsFuncionals: Definició expressions regulars per al dispositiu Receipt
- 4.1.1. TestsFuncionals: Definició expressions regulars per al dispositiu SecureKey
- 4.1.1. TestsFuncionals: Definició expressions regulars per al dispositiu Dispenser
- 4.1.1. TestsFuncionals: Definició expressions regulars per al dispositiu Device
- 4.1.1. TestsFuncionals: Definició expressions regulars per al dispositiu MNT
- 4.1.1. TestsFuncionals: Definició expressions regulars per al dispositiu RKL
- 4.2.1. ContextLoader: Connexió amb el ContextLoaderManager
- 4.2.2. ContextLoader: càrrega en temps d'execució de contextos

4.3.1. SDLController: Desenvolupament d'un controlador remot

Macro tasca 5: Automatització

-Totes les tasques relacionades amb l'automatització de l'entorn de proves.

Tasques :

5.1.1. Automatització: script d'automatització

5.2.1. Validació: execució monitoritzada d'un conjunt de tests

Macro tasca 6: Documentació

-Totes les tasques relacionades amb documentar aquest projecte.

Tasques :

6.1.1. Documentació interna: Documentació tasca 2.1.1. i 2.1.2.

6.1.2. Documentació interna: Documentació tasca 2.1.3. i 2.1.4.

6.1.3. Documentació interna: Documentació tasca 2.1.5. i 2.1.6.

6.1.4. Documentació interna: Documentació tasca 2.1.7. i 2.1.8.

6.1.5. Documentació interna: Documentació tasca 2.1.9. i 2.1.10.

6.1.6. Documentació interna: Documentació tasca 2.1.11. i 2.1.12.

6.1.7. Documentació interna: Documentació tasca 2.1.13. i 2.1.14.

6.1.8. Documentació interna: Documentació tasca 5.1.1.

6.2.1. Documentació acadèmica: Entregues GEP

6.2.2. Documentació acadèmica: Memòria TFG

6.2.3. Documentació acadèmica: Presentació TFG

8.2. Diagrama de Gantt

Hem utilitzat un diagrama de Gantt per mostrar com seria possible assolir aquest *milestone* en les pròximes iteracions. És una planificació possible, en cap cas definitiva, ja que a cada reunió de *sprint planning* caldrà valorar si és adequada aquesta planificació, si les tasques planificades són viables en el període d'un *sprint* i si no hi ha cap prioritització o canvi.

Macro tasca	Tasca	Nov. 2015			Des. 2015			Gen. 2016				Feb. 2016				Mar. 2016				
		9-13	16-20	23-27	30-4	7-11	14-18	21-23	4-8	11-15	18-22	25-29	1-5	8-12	15-19	22-26	29-4	7-11	14-18	21-24
Familiarització	1.1. Familiarització amb l'arquitectura AMT for TSB																			
	1.2. Instal·lació de l'entorn de desenvolupament SDL																			
	1.3. Instal·lació de l'entorn de desenvolupament AppChannel																			
Emulador	2.1.1. Emulador: Desenvolupar l'emulador per al dispositiu Magstripe																			
	2.1.2. Emulador: Test unitaris de l'emulador per al dispositiu Magstripe																			
	2.1.3. Emulador: Desenvolupar l'emulador per al dispositiu Receipt																			
	2.1.4. Emulador: Test unitaris de l'emulador per al dispositiu Receipt																			
	2.1.5. Emulador: Desenvolupar l'emulador per al dispositiu SecureKey																			
	2.1.6. Emulador: Test unitaris de l'emulador per al dispositiu Securekey																			
	2.1.7. Emulador: Desenvolupar l'emulador per al dispositiu Dispenser																			
	2.1.8. Emulador: Test unitaris de l'emulador per al dispositiu Dispenser																			
	2.1.9. Emulador: Desenvolupar l'emulador per al dispositiu Device																			
	2.1.10. Emulador: Test unitaris de l'emulador per al dispositiu Device																			
	2.1.11. Emulador: Desenvolupar l'emulador per al dispositiu MNT																			
	2.1.12. Emulador: Test unitaris de l'emulador per el dispositiu MNT																			
	2.1.13. Emulador: Desenvolupar l'emulador per al dispositiu RKL																			
	2.1.14. Emulador: Test unitaris de l'emulador per al dispositiu RKL																			
Entorn d'execució	2.2.1. ContextLoaderManager: Desenvolupar EndPoint																			
	2.2.2. ContextLoaderManager: Desenvolupar Core (carrega de contextos)																			
	2.2.3. ContextLoaderManager: Desenvolupar Descodificadors																			
	3.1.1. Contexts: Creació dels contextos per al dispositiu Magstripe																			
	3.1.2. Contexts: Creació dels contextos per al dispositiu Receipt																			
	3.1.3. Contexts: Creació dels contextos per al dispositiu SecureKey																			
	3.1.4. Contexts: Creació dels contextos per al dispositiu Dispenser																			
Entorn de prova	3.1.5. Contexts: Creació dels contextos per al dispositiu Device																			
	3.1.6. Contexts: Creació dels contextos per al dispositiu MNT																			
	3.1.7. Contexts: Creació dels contextos per al dispositiu RKL																			
	4.1.1. TestsFuncionals: Definició expressions regulars per al dispositiu Magstripe																			
	4.1.2. TestsFuncionals: Definició expressions regulars per al dispositiu Receipt																			
	4.1.1. TestsFuncionals: Definició expressions regulars per al dispositiu SecureKey																			
	4.1.1. TestsFuncionals: Definició expressions regulars per al dispositiu Dispenser																			
Automatització	4.1.1. TestsFuncionals: Definició expressions regulars per al dispositiu Device																			
	4.1.1. TestsFuncionals: Definició expressions regulars per al dispositiu MNT																			
	4.1.1. TestsFuncionals: Definició expressions regulars per al dispositiu RKL																			
	4.2.1. ContextLoader: Connexió amb el ContextLoaderManager																			
	4.2.2. ContextLoader: Carrega en temps d'execució de contextos																			
	4.3.1. SDLController: Desenvolupament d'un controlador remot																			
	5.1.1. Automatització: script d'automatització																			
Documentació	5.2.1. Validació: execució monitoritzada d'un conjunt de tests																			
	6.1.1. Documentació interna: Documentació tasca 2.1.1. i 2.1.2.																			
	6.1.2. Documentació interna: Documentació tasca 2.1.3. i 2.1.4.																			
	6.1.3. Documentació interna: Documentació tasca 2.1.5. i 2.1.6.																			
	6.1.4. Documentació interna: Documentació tasca 2.1.7. i 2.1.8.																			
	6.1.5. Documentació interna: Documentació tasca 2.1.9. i 2.1.10.																			
	6.1.6. Documentació interna: Documentació tasca 2.1.11. i 2.1.12.																			
	6.1.7. Documentació interna: Documentació tasca 2.1.13. i 2.1.14.																			
	6.1.8. Documentació interna: Documentació tasca 5.1.1.																			
	6.2.1. Documentació acadèmica: Entregues GEP																			
6.2.2. Documentació acadèmica: Memòria TFG																				
6.2.3. Documentació acadèmica: Presentació TFG																				

Il·lustració 9: Diagrama de Gantt part I (de Novembre a Març)

Macro tasca	Tasca	Abr. 2016					Mai. 2016				Jun. 2016				
		29-1	4-8	11-15	18-22	25-29	2-6	9-13	16-20	23-27	30-3	6-10	13-17	20-24	27-30
Familiarització	1.1. Familiarització amb l'arquitectura AMT for TSB														
	1.2. Instal·lació de l'entorn de desenvolupament SDL														
	1.3. Instal·lació de l'entorn de desenvolupament AppChannel														
Emulador	2.1.1. Emulador: Desenvolupar l'emulador per al dispositiu Magstripe														
	2.1.2. Emulador: Test unitaris de l'emulador per al dispositiu Magstripe														
	2.1.3. Emulador: Desenvolupar l'emulador per al dispositiu Receipt														
	2.1.4. Emulador: Test unitaris de l'emulador per al dispositiu Receipt														
	2.1.5. Emulador: Desenvolupar l'emulador per al dispositiu SecureKey														
	2.1.6. Emulador: Test unitaris de l'emulador per al dispositiu SecureKey														
	2.1.7. Emulador: Desenvolupar l'emulador per al dispositiu Dispenser														
	2.1.8. Emulador: Test unitaris de l'emulador per al dispositiu Dispenser														
	2.1.9. Emulador: Desenvolupar l'emulador per al dispositiu Device														
	2.1.10. Emulador: Test unitaris de l'emulador per al dispositiu Device														
	2.1.11. Emulador: Desenvolupar l'emulador per al dispositiu MNT														
	2.1.12. Emulador: Test unitaris de l'emulador per al dispositiu MNT														
	2.1.13. Emulador: Desenvolupar l'emulador per al dispositiu RKL														
	2.1.14. Emulador: Test unitaris de l'emulador per al dispositiu RKL														
Entorn d'execució	2.2.1. ContextLoaderManager: Desenvolupar EndPoint														
	2.2.2. ContextLoaderManager: Desenvolupar Core (carrega de contexts)														
	2.2.3. ContextLoaderManager: Desenvolupar Descodificadors														
Entorn de prova	3.1.1. Contexts: Creació dels contexts per al dispositiu Magstripe														
	3.1.2. Contexts: Creació dels contexts per al dispositiu Receipt														
	3.1.3. Contexts: Creació dels contexts per al dispositiu SecureKey														
	3.1.4. Contexts: Creació dels contexts per al dispositiu Dispenser														
	3.1.5. Contexts: Creació dels contexts per al dispositiu Device														
	3.1.6. Contexts: Creació dels contexts per al dispositiu MNT														
	3.1.7. Contexts: Creació dels contexts per al dispositiu RKL														
Automatització	4.1.1. TestsFuncionals: Definició expressions regulars per al dispositiu Magstripe														
	4.1.2. TestsFuncionals: Definició expressions regulars per al dispositiu Receipt														
	4.1.1. TestsFuncionals: Definició expressions regulars per al dispositiu SecureKey														
	4.1.1. TestsFuncionals: Definició expressions regulars per al dispositiu Dispenser														
	4.1.1. TestsFuncionals: Definició expressions regulars per al dispositiu Device														
	4.1.1. TestsFuncionals: Definició expressions regulars per al dispositiu MNT														
	4.1.1. TestsFuncionals: Definició expressions regulars per al dispositiu RKL														
	4.2.1. ContextLoader: Connexió amb el ContextLoaderManager														
	4.2.2. ContextLoader: Carrega en temps d'execució de contexts														
	4.3.1. SDLController: Desenvolupament d'un controlador remot														
Documentació	5.1.1. Automatització: script d'automatització														
	5.2.1. Validació: execució monitoritzada d'un conjunt de tests														
	6.1.1. Documentació interna: Documentació tasca 2.1.1. i 2.1.2.														
	6.1.2. Documentació interna: Documentació tasca 2.1.3. i 2.1.4.														
	6.1.3. Documentació interna: Documentació tasca 2.1.5. i 2.1.6.														
	6.1.4. Documentació interna: Documentació tasca 2.1.7. i 2.1.8.														
	6.1.5. Documentació interna: Documentació tasca 2.1.9. i 2.1.10.														
	6.1.6. Documentació interna: Documentació tasca 2.1.11. i 2.1.12.														
	6.1.7. Documentació interna: Documentació tasca 2.1.13. i 2.1.14.														
	6.1.8. Documentació interna: Documentació tasca 5.1.1.														
	6.2.1. Documentació acadèmica: Entregues TFG														
	6.2.2. Documentació acadèmica: Memòria TFG														
	6.2.3. Documentació acadèmica: Presentació TFG														

Il·lustració 10: Diagrama de Gantt part II (d'Abril a Juny)

9. Especificació de requisits

9.1. Requisits funcionals

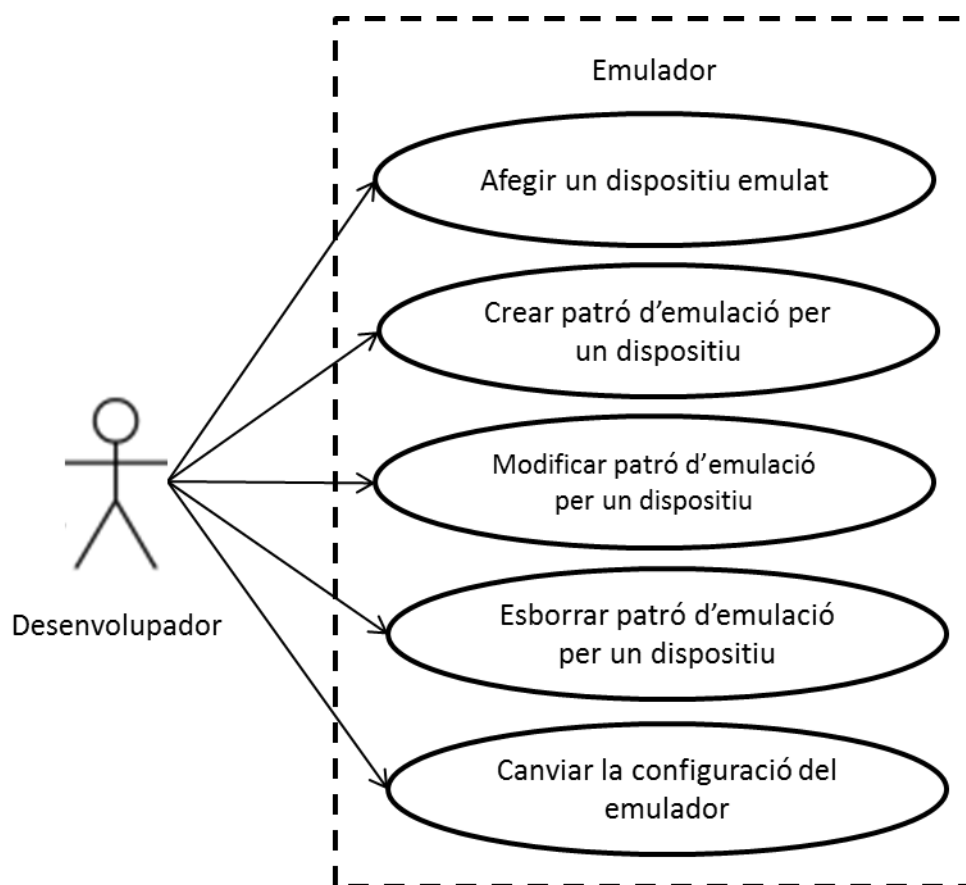
En aquest apartat s'especifiquen els requisits de l'*E2E-framework*, és a dir tot allò que el sistema ha de ser capaç de fer.

Tot seguit, exposo els diagrames dels casos d'ús amb la seva descripció. Després d'aquestes descripcions trobarem el diagrama conceptual de la solució. I per últim la descripció dels requisits no funcionals d'aquest sistema.

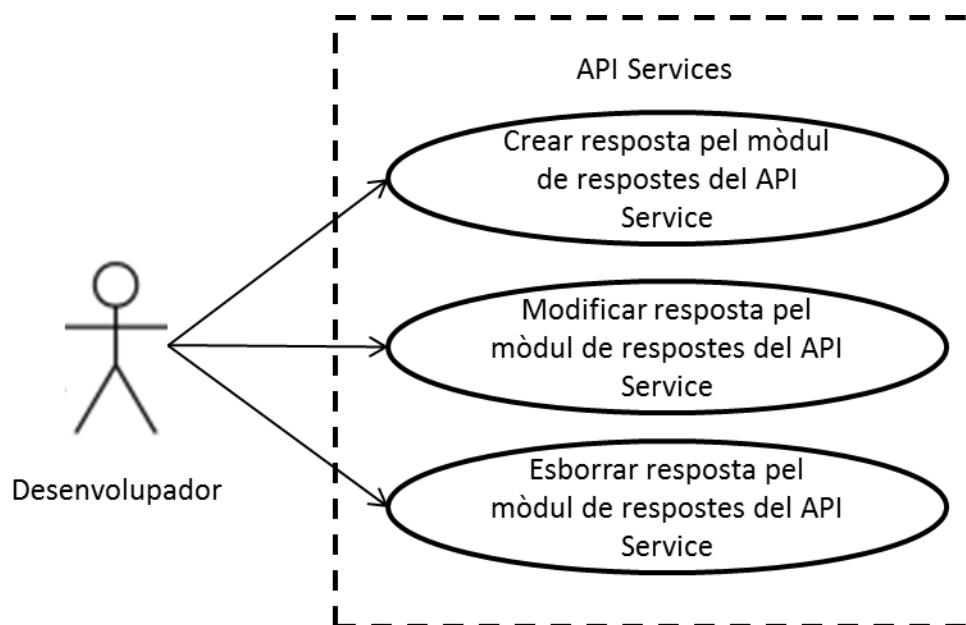
9.1.1. Diagrama de casos d'ús

En primer lloc presentem els diagrames dels casos d'ús dels components

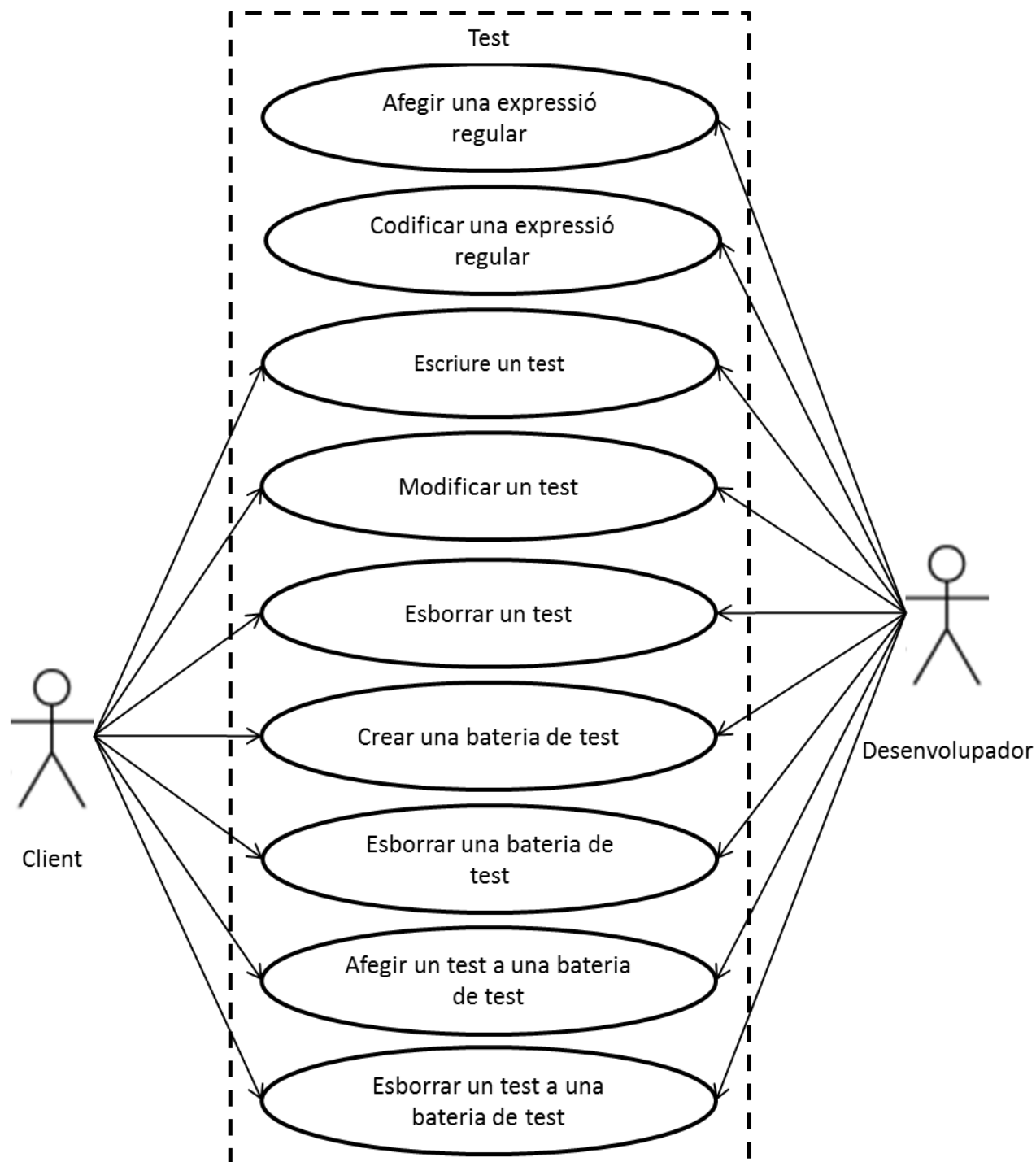
- Emulador (Il·lustració 11)
- Mòdul de respostes del API Service (Il·lustració 12)
- Tests (Il·lustració 13)
- La resta de casos d'ús referents al *framework*, és a dir, casos d'ús a nivell de sistema complet on intervenen tots els components (il·lustració 14).



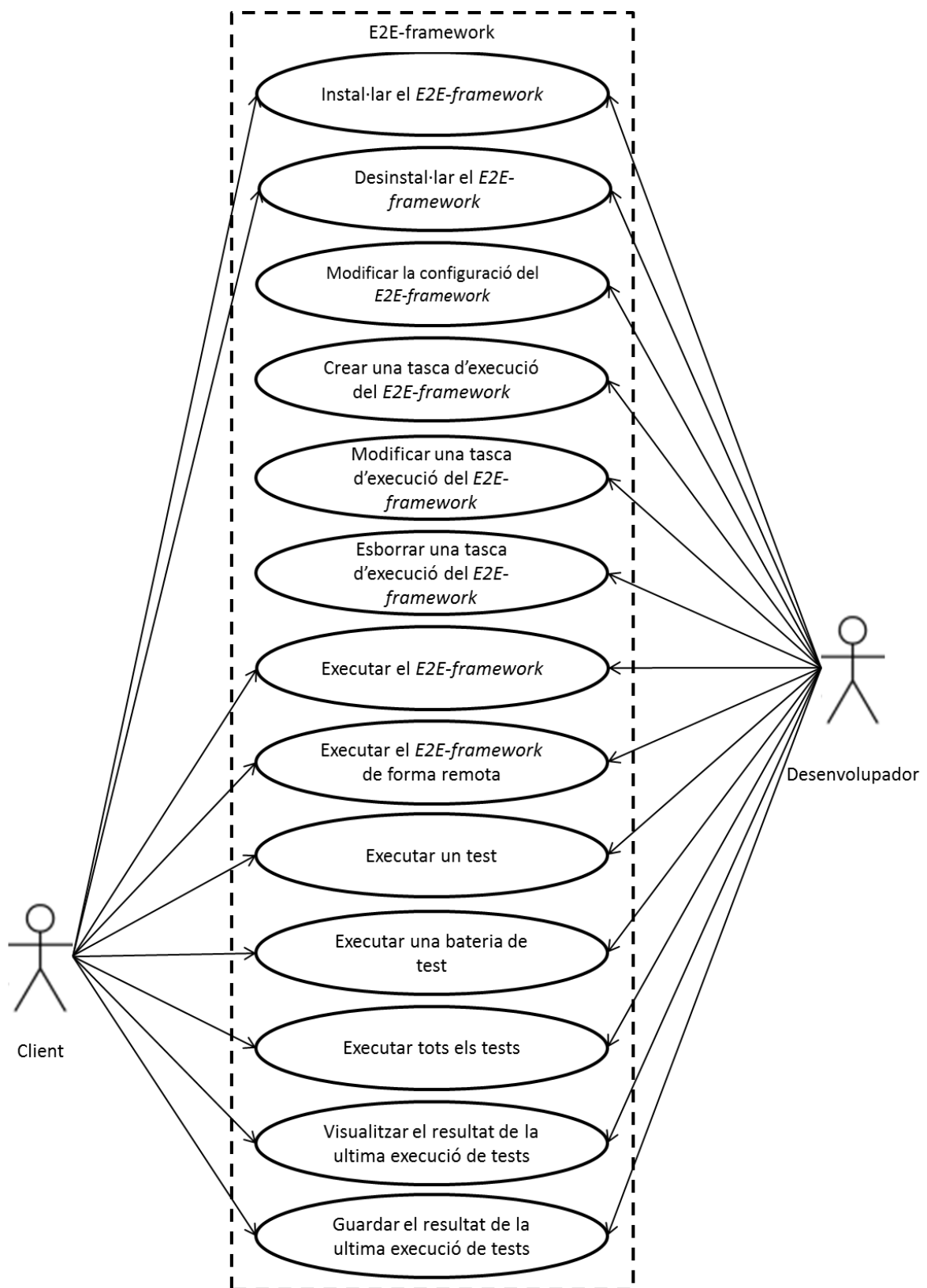
Il·lustració 11: Diagrama de casos d'ús de l'Emulador



Il·lustració 12: Diagrama de casos d'us de l'API Service



Il·lustració 13: Diagrama de casos d'ús dels Test



Il·lustració 14: Diagrama de casos d'us del E3E-framework

9.1.2. Descripció dels casos d'ús

En aquest apartat es fa la descripció de cadascun dels casos d'ús. Per descriure'ls utilitzo la següent plantilla:

ID	ID del cas d'ús
Nom	Nom del cas d'ús
Actors principals	Usuari que fa ús d'aquesta funcionalitat
Precondició	El que s'ha de complir abans d'executar el cas d'ús
Activador	El que condiciona que s'executi aquest cas d'ús
Escenari principal	Comportament habitual del cas d'ús
Extincions	Comportament alternatiu del cas d'ús

ID	CU01
Nom	Afegir un dispositiu emulat
Actors principals	Desenvolupador
Precondició	Es disposa del codi de l'emulador i el dispositiu a afegir no existeix
Activador	Necessitat d'emular un dispositiu físic
Escenari principal	1. L'actor escriu els fitxers per emular el dispositiu 2. L'actor afegeix el codi del nou dispositiu 3. El sistema comprova que els tests unitaris de l'emulador se superen satisfactòriament 4. El sistema compila l'emulador 5. L'actor documenta la seva actuació 6. L'actor publica el codi per la resta del equip
Extincions	3a. Els test unitaris de l'emulador no passen 3a1. S'acaba el cas d'ús 4a. La compilació falla 4a1. S'acaba el cas d'ús

ID	CU02
Nom	Crear un patró d'emulació per un dispositiu
Actors principals	Desenvolupador
Precondició	Es disposa del codi de l'emulador i el dispositiu existeix en el codi i aquest està aturat
Activador	Es vol incorporar una casuística nova per a aquest dispositiu
Escenari principal	1. L'actor escriu el patró d'emulació 2. L'actor afegeix el fitxer del patró a l'emulador 3. El sistema compila l'emulador 4. L'actor prova l'emulador amb aquest nou patró 5. L'actor documenta l'actuació 6. L'actor publica el codi per la resta del equip
Extincions	3a. Els test unitaris de l'emulador no passen 3a1. S'acaba el cas d'ús 4a. La compilació falla 4a1. S'acaba el cas d'ús

ID	CU03
Nom	Modificar un patró d'emulació per un dispositiu
Actors principals	Desenvolupador
Precondició	Es disposa del codi del emulador, existeix el dispositiu i el fitxer que conté el patró a modificar i l'emulador està aturat
Activador	Es vol canviar el patró que ja està escrit
Escenari principal	<ol style="list-style-type: none"> 1. L'actor localitza el fitxer a modificar 2. L'actor escriu les modificacions 3. El sistema comprova que els tests unitaris de l'emulador se superen satisfactòriament 4. El sistema compila l'emulador 5. L'actor modifica la documentació 6. L'actor publica el codi per la resta del equip
Extincions	3a. Els test unitaris de l'emulador no passen 3a1. S'acaba el cas d'ús 4a. La compilació falla 4a1. S'acaba el cas d'ús

ID	CU04
Nom	Esborrar un patró d'emulació per un dispositiu
Actors principals	Desenvolupador
Precondició	Es disposa del codi de l'emulador, existeix com a mínim un fitxer que conté un patró d'emulació i l'emulador està aturat
Activador	Es vol eliminar un patró d'emulació
Escenari principal	<ol style="list-style-type: none"> 1. L'actor localitza el fitxer a esborrar 2. L'actor esborra el fitxer 3. El sistema compila l'emulador 4. L'actor esborra la documentació 5. L'actor publica el codi per la resta del equip
Extincions	3a. La compilació falla 3a1. S'acaba el cas d'ús

ID	CU05
Nom	Canviar la configuració de l'emulador
Actors principals	Desenvolupador
Precondició	Es disposa del codi i la configuració actual de l'emulador i aquest està aturat
Activador	Es vol canviar la configuració de l'emulador
Escenari principal	<ol style="list-style-type: none"> 1. L'actor localitza els fitxers de configuració 2. L'actor modifica la configuració 3. El sistema funciona amb la nova configuració de l'emulador 4. L'actor documenta la seva actuació 5. L'actor publica el codi per la resta del equip
Extincions	3a. El sistema falla 3a1. S'acaba el cas d'ús

ID	CU06
Nom	Crear una resposta per al mòdul de respostes del API Service
Actors principals	Desenvolupador
Precondició	Es disposa del codi del mòdul de respostes i aquest està aturat
Activador	Es vol afegir una nova resposta
Escenari principal	<ol style="list-style-type: none"> 1. L'actor escriu la resposta en el fitxer de respostes 2. L'actor posa en marxa el mòdul 3. El sistema consumeix aquesta resposta 4. L'actor comprova que la resposta s'envia correctament 5. L'actor documenta la seva actuació 6. L'actor publica el codi per la resta del equip
Extincions	<ol style="list-style-type: none"> 3a. El sistema no pot consumir la resposta 3a1. S'acaba el cas d'ús 4a. Les proves fallen 4a1. S'acaba el cas d'ús

ID	CU07
Nom	Modificar una resposta per al mòdul de respostes del API Service
Actors principals	Desenvolupador
Precondició	Es disposa del codi del mòdul de respostes, existeix el fitxer que conté els respostes amb una resposta com a mínim i aquest mòdul està aturat
Activador	Es vol canviar una resposta
Escenari principal	<ol style="list-style-type: none"> 1. L'actor localitza el fitxer a modificar 2. L'actor escriu les modificacions 3. L'actor posa en marxa el mòdul 4. El sistema consumeix aquesta resposta 5. L'actor comprova que la resposta s'envia correctament 6. L'actor documenta la seva actuació 7. L'actor publica el codi per la resta del equip
Extincions	<ol style="list-style-type: none"> 4a. El sistema no pot consumir la resposta 4a1. S'acaba el cas d'ús 5a. Les proves fallen 5a1. S'acaba el cas d'ús

ID	CU08
Nom	Esborrar una resposta per al mòdul de respostes del API Service
Actors principals	Desenvolupador
Precondició	Es disposa del codi del mòdul de respostes, existeix com a mínim una resposta en el fitxer que conte els respostes i aquest el mòdul està aturat
Activador	Es vol esborrar una resposta
Escenari principal	<ol style="list-style-type: none"> 1. L'actor localitza el fitxer a esborrar 2. L'actor esborra el fitxer 3. El sistema ja no disposa del fitxer 4. L'actor documenta la seva actuació 5. L'actor publica el codi per la resta del equip
Extincions	3a. El sistema disposa del fitxer 3a1. Passem al pas 1 del cas d'ús

ID	CU09
Nom	Afegir una expressió regular
Actors principals	Desenvolupador
Precondició	Existeix el fitxer d'expressions regulars i l'expressió a afegir no existeix
Activador	Es necessita una nova expressió regular
Escenari principal	<ol style="list-style-type: none"> 1. L'actor localitza el fitxer d'expressions regulars 2. L'actor defineix l'expressió regular 3. El sistema identifica l'expressió regular 4. L'actor publica el fitxer per la resta del equip
Extincions	3a. El sistema no pot identificar l'expressió regular 3a1. S'acaba el cas d'ús

ID	CU10
Nom	Codificar una expressió regular
Actors principals	Desenvolupador
Precondició	Existeix el fitxer d'expressions regulars i l'expressió no està codificada
Activador	Es vol codificar una expressió regular
Escenari principal	<ol style="list-style-type: none"> 1. L'actor localitza el fitxer d'expressions regulars 2. L'actor codifica l'expressió regular 3. El sistema pot executar la codificació de l'expressió regular 4. L'actor publica el fitxer per la resta del equip
Extincions	3a. El sistema no pot executar la codificació 3a1. S'acaba el cas d'ús

ID	CU11
Nom	Escriure un test
Actors principals	Client o desenvolupador
Precondició	Existeix el fitxer d'expressions regulars i existeix la carpeta de tests
Activador	Es vol escriure un nou test funcional
Escenari principal	<ol style="list-style-type: none"> 1. L'actor escriu el test 2. L'actor afegeix el test a la carpeta de tests 3. El sistema identifica el fitxer com un test 4. L'actor documenta la seva actuació
Extincions	3a. El sistema no pot identificar el fitxer 3a1. S'acaba el cas d'ús

ID	CU12
Nom	Modificar un test
Actors principals	Client o desenvolupador
Precondició	Existeix el fitxer d'expressions regulars, existeix la carpeta de tests i existeix un test com a mínim
Activador	Es vol modificar un test funcional
Escenari principal	<ol style="list-style-type: none"> 1. L'actor localitza el test a modificar 2. L'actor realitza les modificacions 3. El sistema identifica el fitxer com un test 4. L'actor documenta la seva actuació
Extincions	3a. El sistema no pot identificar el fitxer 3a1. S'acaba el cas d'ús

ID	CU13
Nom	Esborrar un test
Actors principals	Client o desenvolupador
Precondició	Existeix el fitxer d'expressions regulars, la carpeta de tests i aquesta conté com a mínim un test
Activador	Es vol esborrar un test funcional
Escenari principal	<ol style="list-style-type: none"> 1. L'actor localitza el test a esborrar 2. L'actor esborra el test 3. El sistema ja no disposa del fitxer 4. L'actor documenta la seva actuació
Extincions	3a. El sistema disposa del fitxer 3a1. S'acaba el cas d'ús

ID	CU14
Nom	Crear una bateria de tests
Actors principals	Client o desenvolupador
Precondició	Existeix el fitxer d'expressions regulars, la carpeta de tests i els tests
Activador	Es vol crear una bateria de test funcionals
Escenari principal	<ol style="list-style-type: none"> 1. L'actor crea una subcarpeta per la bateria 2. L'actor localitza els test que vol afegir a la bateria 3. L'actor mou els test a la subcarpeta 4. El sistema identifica la nova carpeta com una bateria de test 5. L'actor documenta la seva actuació
Extincions	<p>4a. El sistema no identifica la carpeta</p> <p>4a1. S'acaba el cas d'ús</p>

ID	CU15
Nom	Esborrar una bateria de tests
Actors principals	Client o desenvolupador
Precondició	Existeix com a mínim una subcarpeta dins la carpeta de test
Activador	Es vol esborrar una bateria de test funcionals
Escenari principal	<ol style="list-style-type: none"> 1. L'actor localitza la bateria de test que vol esborrar 2. L'actor esborra la subcarpeta 4. El sistema esborra la bateria de tests 3. L'actor documenta la seva actuació
Extincions	<p>4a. El sistema no pot esborrar la bateria</p> <p>4a1. S'acaba el cas d'ús</p>

ID	CU16
Nom	Afegir un test a una bateria de tests
Actors principals	Client o desenvolupador
Precondició	Existeix la carpeta de tests i la subcarpeta de la bateria
Activador	Es vol afegir un test funcional a una bateria en concret
Escenari principal	<ol style="list-style-type: none"> 1. L'actor localitza el test que vol afegir 2. L'actor mou el test a la bateria de test 3. El sistema afegeix el test a la bateria 4. L'actor documenta la seva actuació
Extincions	<p>3a. El sistema no pot afegir el test a la bateria de tests</p> <p>3a1. S'acaba el cas d'ús</p>

ID	CU17
Nom	Esborrar un test a una bateria de test
Actors principals	Client o desenvolupador
Precondició	Existeix la carpeta de tests, la subcarpeta de la bateria i aquesta conté tests
Activador	Es vol esborrar un test funcional a una bateria en concret
Escenari principal	<ol style="list-style-type: none"> 1. L'actor localitza el test que vol esborrar 2. L'actor mou el test fora de la subcarpeta de la bateria de tests 4. El sistema esborra el test de la bateria 3. L'actor documenta la seva actuació
Extincions	3a. El sistema no pot esborrar el test de la bateria de tests 3a1. S'acaba el cas d'ús

ID	CU18
Nom	Instal·lar l' <i>E2E-framework</i>
Actors principals	Client o desenvolupador
Precondició	Es disposa d'un ordinador, la guia d'instal·lació, els fitxers que conformen l' <i>E2E-framework</i> i l'instal·lador
Activador	Es vol fer ús del l' <i>E2E-framework</i>
Escenari principal	<ol style="list-style-type: none"> 1. L'actor llegeix la guia d'instal·lació 2. L'actor desempaqueta els components de l'<i>E2E-framework</i> 3. L'actor executa l'instal·lador 4. El sistema descarrega les dependències externes 5. El sistema s'instal·la
Extincions	1a. La guia no està actualitzada 1a1. S'acaba el cas d'ús 4a. El sistema no ha pogut descarregar les dependències 4a1. S'acaba el cas d'ús 5a. Falla la instal·lació 5a1. S'acaba el cas d'ús

ID	CU19
Nom	Desinstal·lar l' <i>E2E-framework</i>
Actors principals	Client o desenvolupador
Precondició	L' <i>E2E-framework</i> està instal·lat
Activador	Es vol eliminar l' <i>E2E-framework</i> de l'ordinador
Escenari principal	<ol style="list-style-type: none"> 1. L'actor obre un terminal de comandes 2. L'actor aturar l'<i>E2E-framework</i> 3. El sistema acaba la seva execució 4. Localitzar on es va fer la instal·lació 5. Esborrar tots els fitxers de l'<i>E2E-framework</i>
Extincions	3b. L' <i>E2E-framework</i> està aturat 3b1. Passem al pas 4

ID	CU20
Nom	Modificar la configuració de l' <i>E2E-framework</i>
Actors principals	Desenvolupador
Precondició	Es disposa dels components que conformen l' <i>E2E-framework</i> i estan instal·lats
Activador	Es vol canviar algun aspecte de configuració de l' <i>E2E-framework</i>
Escenari principal	<ol style="list-style-type: none"> 1. L'actor obre un terminal de comandes 2. L'actor aturar l'<i>E2E-framework</i> 3. El sistema acaba la seva execució 4. Localitzar la configuració 5. Realitzar les modificacions
Extincions	3b. L' <i>E2E-framework</i> està aturat 3b1. Passem al pas 3

ID	CU21
Nom	Crear una tasca d'execució de l' <i>E2E-framework</i>
Actors principals	Desenvolupador
Precondició	Es disposa dels components que conformen l' <i>E2E-framework</i> i estan instal·lats
Activador	Es vol una tasca d'execució
Escenari principal	<ol style="list-style-type: none"> 1. L'actor obre un terminal de comandes 2. L'actor aturar l'<i>E2E-framework</i> 3. El sistema acaba la seva execució 4. Localitzar el fitxer que conte les tasques 5. Afegir la tasca
Extincions	3b. L' <i>E2E-framework</i> està aturat 3b1. Passem al pas 3

ID	CU22
Nom	Modificar una tasca d'execució de l' <i>E2E-framework</i>
Actors principals	Desenvolupador
Precondició	Es disposa dels components que conformen l' <i>E2E-framework</i> , estan instal·lats i existeix una tasca com a mínim
Activador	Es vol modificar una tasca d'execució
Escenari principal	<ol style="list-style-type: none"> 1. L'actor obre un terminal de comandes 2. L'actor aturar l'<i>E2E-framework</i> 3. El sistema acaba la seva execució 4. Localitzar el fitxer que conte les tasques 5. Modificar la tasca
Extincions	3b. L' <i>E2E-framework</i> està aturat 3b1. Passem al pas 3

ID	CU23
Nom	Esborrar una tasca d'execució de l' <i>E2E-framework</i>
Actors principals	Desenvolupador
Precondició	Es disposa dels components que conformen l' <i>E2E-framework</i> , estan instal·lats i existeix una tasca com a mínim
Activador	Es vol esborrar una tasca d'execució
Escenari principal	<ol style="list-style-type: none"> 1. L'actor obre un terminal de comandes 2. L'actor aturar l'<i>E2E-framework</i> 3. El sistema acaba la seva execució 4. Localitzar el fitxer que conté les tasques 5. Esborrar la tasca
Extincions	3b. L' <i>E2E-framework</i> està aturat 3b1. Passem al pas 3

ID	CU24
Nom	Executar l' <i>E2E-framework</i>
Actors principals	Client i desenvolupador
Precondició	Es disposa dels components que conformen l' <i>E2E-framework</i> , estan instal·lats i existeix una tasca d'execució com a mínim
Activador	Es vol executar l' <i>E2E-framework</i>
Escenari principal	<ol style="list-style-type: none"> 1. L'actor obre un terminal de comandes 2. L'actor escriu la comanda per inicialitzar l'<i>E2E-framework</i> 3. L'actor executa la comanda 4. El sistema interpreta la comanda 5. El sistema executa la tasca
Extincions	4a. La comanda no existeix o està mal escrita 4a1. S'acaba el cas d'ús 5a. L' <i>E2E-framework</i> està en execució 5a1. S'acaba el cas d'ús

ID	CU25
Nom	Executar l' <i>E2E-framework</i> de forma remota
Actors principals	Client i desenvolupador
Precondició	Es disposa dels components que conformen l' <i>E2E-framework</i> , estan instal·lats, existeix una tasca d'execució com a mínim i es disposa d'accés a la màquina de forma remota
Activador	Es vol executar l' <i>E2E-framework</i> de forma remota
Escenari principal	<ol style="list-style-type: none"> 1. L'actor obre un terminal de comandes 2. L'actor accedeix de forma remota a la màquina que on hi ha l'<i>E2E-framework</i> 3. L'actor escriu la comanda per inicialitzar l'<i>E2E-framework</i> 4. L'actor executa la comanda 5. El sistema interpreta la comanda 6. El sistema executa la tasca
Extincions	<p>5a. La comanda no existeix o està mal escrita</p> <p>5a1. S'acaba el cas d'ús</p> <p>6a. L'<i>E2E-framework</i> està en execució</p> <p>6a1. S'acaba el cas d'ús</p>

ID	CU26
Nom	Executar un test
Actors principals	Client i desenvolupador
Precondició	Es disposa dels components que conformen l' <i>E2E-framework</i> , estan instal·lats, existeix una tasca d'execució com a mínim
Activador	Es vol executar un test funcional amb l' <i>E2E-framework</i>
Escenari principal	<ol style="list-style-type: none"> 1. L'actor obre un terminal de comandes 2. L'actor escriu la tasca que indica a l'<i>E2E-framework</i> els tests a executar 3. L'actor executa la comanda 4. El sistema interpreta la comanda 5. El sistema executa la tasca
Extincions	<p>4a. La comanda no existeix o està mal escrita</p> <p>4a1. S'acaba el cas d'ús</p> <p>5a. L'<i>E2E-framework</i> està en execució</p> <p>5a1. S'acaba el cas d'ús</p> <p>5b. L'<i>E2E-framework</i> no completa l'execució correctament</p> <p>5b1. Mostra l'error més plausible</p> <p>5b2. S'acaba el cas d'ús</p>

ID	CU27
Nom	Executar una bateria de tests
Actors principals	Client i desenvolupador
Precondició	Es disposa dels components que conformen l' <i>E2E-framework</i> , estan instal·lats, existeix una tasca d'execució com a mínim
Activador	Es vol executar d'una bateria de tests funcional amb l' <i>E2E-framework</i>
Escenari principal	<ol style="list-style-type: none"> 1. L'actor obre un terminal de comandes 2. L'actor escriu la tasca que indica al <i>E2E-framework</i> la bateria a executar 3. L'actor executa la comanda 4. El sistema interpreta la comanda 5. El sistema executa la tasca
Extincions	<ol style="list-style-type: none"> 4a. La comanda no existeix o està mal escrita <ol style="list-style-type: none"> 4a1. S'acaba el cas d'ús 5a. L'<i>E2E-framework</i> està en execució <ol style="list-style-type: none"> 5a1. S'acaba el cas d'ús 5b. L'<i>E2E-framework</i> no completa l'execució correctament <ol style="list-style-type: none"> 5b1. Mostra l'error més plausible 5b2. S'acaba el cas d'ús

ID	CU28
Nom	Executar tots els tests
Actors principals	Client i desenvolupador
Precondició	Es disposa dels components que conformen l' <i>E2E-framework</i> , estan instal·lats, existeix una tasca d'execució com a mínim
Activador	Es vol executar tots els tests funcional amb l' <i>E2E-framework</i>
Escenari principal	<ol style="list-style-type: none"> 1. L'actor obre un terminal de comandes 2. L'actor escriu la tasca que indica a l'<i>E2E-framework</i> que executi tots els tests 3. L'actor executa la comanda 4. El sistema interpreta la comanda 5. El sistema executa la tasca
Extincions	<ol style="list-style-type: none"> 4a. La comanda no existeix o està mal escrita <ol style="list-style-type: none"> 4a1. S'acaba el cas d'ús 5a. L'<i>E2E-framework</i> està en execució <ol style="list-style-type: none"> 5a1. S'acaba el cas d'ús 5b. L'<i>E2E-framework</i> no completa l'execució correctament <ol style="list-style-type: none"> 5b1. Mostra l'error més plausible 5b2. S'acaba el cas d'ús

ID	CU29
Nom	Visualitzar el resultat de l'última execució de tests
Actors principals	Client i desenvolupador
Precondició	Es disposa dels components que conformen l' <i>E2E-framework</i> , estan instal·lats i existeix una tasca d'execució com a mínim
Activador	Es vol visualitzar els resultat de l'última execució
Escenari principal	1. L' <i>E2E-framework</i> ha finalitzat una execució correctament (cas d'ús: 26, 27, 28) 2. L' <i>E2E-framework</i> ha generat el resultat 3. L'actor visualitza el resultat
Extincions	2a. L' <i>E2E-framework</i> no ha generat el resultat 2a1. Mostra l'error més plausible 2a2. S'acaba el cas d'ús

ID	CU30
Nom	Guardar el resultat de la ultima execució de tests
Actors principals	Client i desenvolupador
Precondició	Es disposa dels components que conformen l' <i>E2E-framework</i> , estan instal·lats i existeix una tasca d'execució com a mínim
Activador	Es vol guardar els resultats de l'última execució
Escenari principal	1. L' <i>E2E-framework</i> ha finalitzat una execució correctament (cas d'ús: 26, 27, 28) 2. L' <i>E2E-framework</i> ha generat el resultat 3. L'actor guarda els resultats
Extincions	2a. L' <i>E2E-framework</i> no ha generat el resultat 2a1. Mostra l'error més plausible 2a2. S'acaba el cas d'ús

9.1.3. Diagrama conceptual

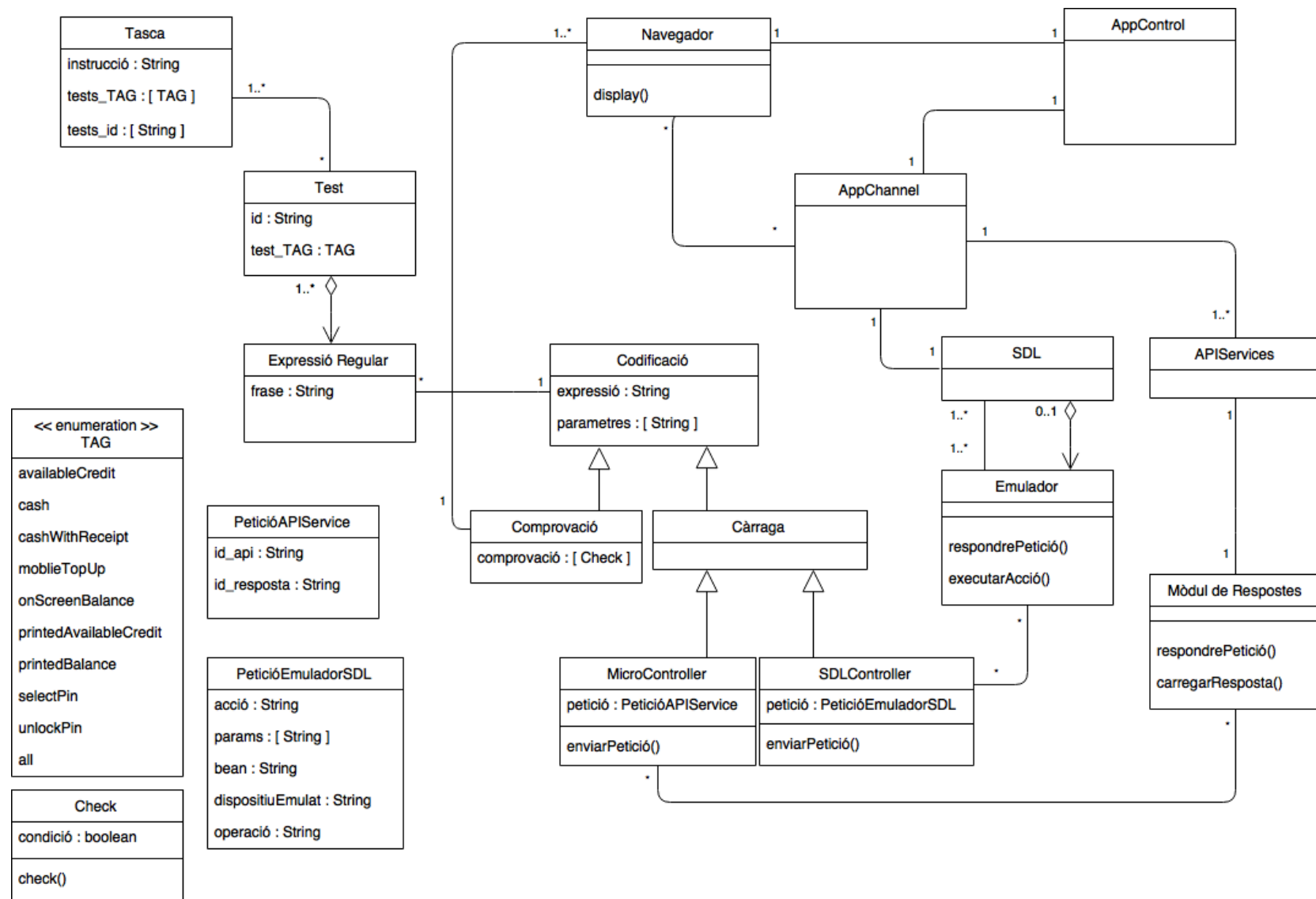
Tot seguit es mostra el diagrama conceptual de l'*E2E-framework* (il·lustració 15), on s'aprecien les diferents entitats que el conformen i les relacions entre elles.

Cal mencionar que SDL, APIService, AppChannel i AppControl es mostren com a entitats que representen un sistema complex el qual té el seu propi diagrama conceptual. És per això que en el diagrama els atributs i operacions d'aquestes entitats estan en blanc. Això és així per dos motius: el primer per simplificar el diagrama i el segon perquè aquest projecte (l'*E2E-framework*) no implementa res en aquestes entitats però si que hi interacciona, motiu pel qual apareixen al diagrama.

Abans del passar al diagrama introduiré les entitats que hi podrem visualitzar. S'introdueixen les entitats perquè, com s'ha mencionat, en el diagrama apareixeran entitats en blanc i d'aquesta forma podem donar-li sentit a l'entitat i aclarir el que està representat en el diagrama.

- Tasca: representa les tasques que el sistema executarà. Una tasca es compon d'una instrucció (per exemple: executar o parar) i dos llistes, la primera és una llista amb els *tags* a executar i la segona una llista de identificadors de test a executar.
- Test: representa com s'identifica i com es compon un test. Un test s'identificarà partir d'un identificador únic i a més a més tindrà un *tag* que compartirà amb més tests. Per últim un test es compon d'expressions regulars.
- Expressió regular: representa una línia d'un test.
- Codificació: representa la generalització d'una codificació d'una expressió regular. Però la funcionalitat d'aquesta codificació pertany a una de les següents especialitzacions
 - Comprovació: representa el conjunt de codificacions especialitzades en fer comprovacions. Totes aquestes estan formades per un llista de condicions a comprovar contra el navegador.
 - Càrrega: representa el conjunt generalitzat de codificacions que tenen com a objectiu carregar dades a l'emulador o al mòdul de respostes.
 - SDLController: representa el conjunt de codificacions especialitzades en fer càrregues a l'emulador. Totes elles realitzen peticions de càrrega de patrons a l'emulador.
 - MicroController: representa el conjunt de codificacions especialitzades en fer càrregues al mòdul de respostes. Totes elles realitzen peticions de càrrega de respostes al mòdul de respostes.
- Emulador: representa l'emulador de dispositius físics del SDL i resol les peticions del SDLController. Aquesta entitat està formada per tots els dispositius emulats que proporcionen dades emulades al SDL.
- Mòdul de respostes: representa el mòdul de respostes del APIService. Aquest entitat està formada pel mòdul que proporciona respostes a l'APIService i resol les peticions del MicroController.
- SDL: representa el component desenvolupat per *ATM for TSB* que s'encarrega d'encapsular les peticions sobre dispositius hardware del caixer (prèviament explicat en l'apartat "2. Context").
- APIService: representa el component desenvolupat per *ATM for TSB* que s'encarrega de gestionar les peticions dels caixers cap als backends del banc (prèviament explicat en l'apartat "2. Context").

- AppControl: representa el component desenvolupat per *ATM for TSB* que s'encarrega gestionar que tots els components estiguin actius durant una transacció (prèviament explicat en l'apartat "2. Context").
- AppChannel: representa el component desenvolupat per *ATM for TSB* que s'encarrega de recuperar les vistes i menús d'opcions de l'APIServices.
- Navegador: representa el navegador que veuen els usuaris en un caixer. On es presenten les vistes i menús d'opcions.



Il·lustració 15: Diagrama conceptual

9.2. Requisits no funcionals

Per descriure els requisits no funcionals utilitzaré la següent plantilla

Requisit #	Numeració del requisit
Descripció	Descripció del requisit
Justificació	Justificació del requisit
Condicció de satisfacció	Condicció objectiva que ha de satisfer el sistema per tal que es pugui considerar que el requisit es compleix

9.2.1. Requisits de *look and feel*

Requisit #	1
Descripció	La sortida visual que produeix el sistema és atractiva i entenedora per als usuaris
Justificació	Cal facilitar als usuaris el consum i interpretació de la informació que generi el sistema
Condicció de satisfacció	Auto-explicatiu

Requisit #	2
Descripció	El disseny de la documentació i altres documents del sistema han de transmetre serietat i professionalitat
Justificació	Cal preservar els valors de serietat i professionalitat
Condicció de satisfacció	El director certificarà que el disseny està alineat amb els valors de l'empresa

9.2.2. Requisits d'usabilitat i humanitat

Requisit #	3
Descripció	Encara que l'usuari no conegui la tecnologia usada per desenvolupar l' <i>E2E-framework</i> serà capaç d'usar-lo
Justificació	El sistema ha d'estar a l'abast de tots els <i>stakeholders</i>
Condicció de satisfacció	Provar el sistema amb <i>stakeholders</i> que no siguin desenvolupadors

Requisit #	4
Descripció	Aprendre a usar l' <i>E2E-framework</i> no serà difícil
Justificació	Cal facilitar l'aprenentatge als usuaris del sistema
Condicció de satisfacció	Amb una petita introducció per part d'un expert l'usuari sabrà tot el que necessita per usar el sistema

Requisit #	5
Descripció	El sistema ha d'intentar evitar els errors d'usuari
Justificació	Donar informació dels possibles errors ajudarà a no prendre decisions errònies per part de l'usuari
Condicció de satisfacció	El sistema informará que l'entrada que dona l'usuari és errònia i proposarà una possible entrada correcta

9.2.3. Requisits de rendiment

Requisit #	6
Descripció	L' <i>E2E-framework</i> ha de ser ràpid
Justificació	Cal disposar del resultat dels tests al més aviat possible
Condicció de satisfacció	Un test no tardarà més de 3 min en passar

Requisit #	7
Descripció	L' <i>E2E-framework</i> ha poder escalar
Justificació	Poder passar més tests
Condicció de satisfacció	El sistema adaptarà el seu rendiment

9.2.4. Requisits operacionals i ambientals

Requisit #	8
Descripció	L' <i>E2E-framework</i> ha de funcionar en Windows i Linux
Justificació	Poder triar el sistema operatiu
Condicció de satisfacció	Auto-explicatiu

Requisit #	9
Descripció	Modificacions i actualitzacions es realitzaran sota petició
Justificació	Adaptar el sistema al que el client demani
Condicció de satisfacció	Auto-explicatiu

9.2.5. Requisits de manteniment i suport

Requisit #	10
Descripció	Existeix un document amb tota la documentació tècnica
Justificació	Document tècnic explicant l' <i>E2E-framework</i>
Condicció de satisfacció	Auto-explicatiu

Requisit #	11
Descripció	Manual d'usuari
Justificació	Document on s'explica com usar l' <i>E2E-framework</i> a nivell d'usuari
Condició de satisfacció	Auto-explicatiu

Requisit #	12
Descripció	Des de la detecció d'un error no tardarem més d'un mes a corregir-lo
Justificació	Cal assegurar el funcionament del sistema i que cap altra funcionalitat ha quedat afectada
Condició de satisfacció	S'ha corregit l'error detectat

Requisit #	13
Descripció	Es podrà consultar els resultats de l' <i>E2E-framework</i> des de Chrome, Firefox i Explorer
Justificació	Tots els usuaris han de poder accedir al sistema
Condició de satisfacció	El sistema presenta els resultats en els navegadors esmentats anteriorment

9.2.6. Requisits de seguretat

Requisit #	14
Descripció	Els components de sistema són segurs
Justificació	El sistema ha de ser fiable: no poden interferir agents desconeguts
Condició de satisfacció	El sistema és segur i els components han estat certificats

Requisit #	15
Descripció	El sistema no serà públic
Justificació	El sistema conté informació sensible i crítica
Condició de satisfacció	El sistema no serà d'accés lliure a tothom

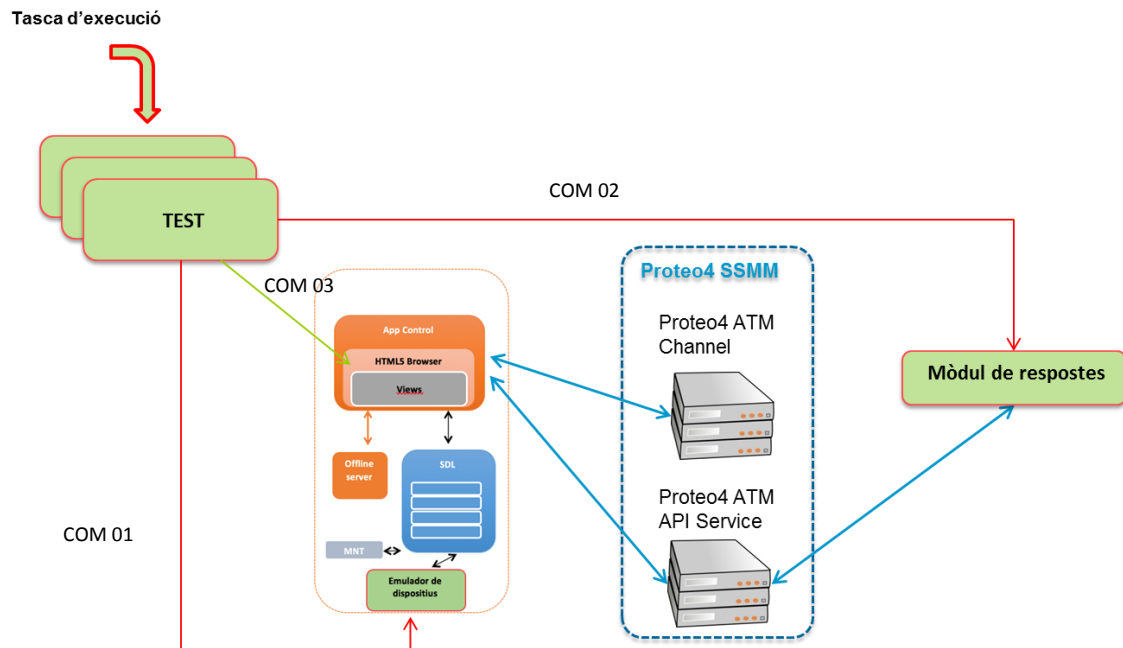
9.2.7. Requisits culturals

Requisit #	16
Descripció	El sistema estarà adaptat a la cultura del client (en aquest cas l'Anglesa)
Justificació	El client acceptarà i validarà el sistema
Condicció de satisfacció	Auto-explicatiu

Requisit #	17
Descripció	El sistema permetrà canviar la informació que presenta per a usuaris de cultures diferents
Justificació	Cal que usuaris de cultures molt diferents se sentin igual de còmodes utilitzant l' <i>E2E-framework</i>
Condicció de satisfacció	Usuaris amb cultures diferents entendran i podran fer ús del sistema

10. Arquitectura de l'E2E-framework

En aquest apartat detallarem els components que conformen el *framework* i el seu funcionament. El punt de partida és la il·lustració 7 on identificàvem certs entorns i components claus. A continuació usem aquesta mateixa il·lustració de base per mostrar les comunicacions entre components.



Il·lustració 16: Diagrama de les comunicacions entre components del E2E-framework

Amb aquesta visió global dels components i les comunicacions etiquetes "COM 0x" que hi ha entre components passarem a veure'l detalladament: començarem per l'emulador del SDL, tot seguit el mòdul de càrrega de respostes de l'API Service i per últim l'arquitectura de tests.

10.1. Emulador

En aquest apartat entrarem en la implementació de l'emulador de dispositius.

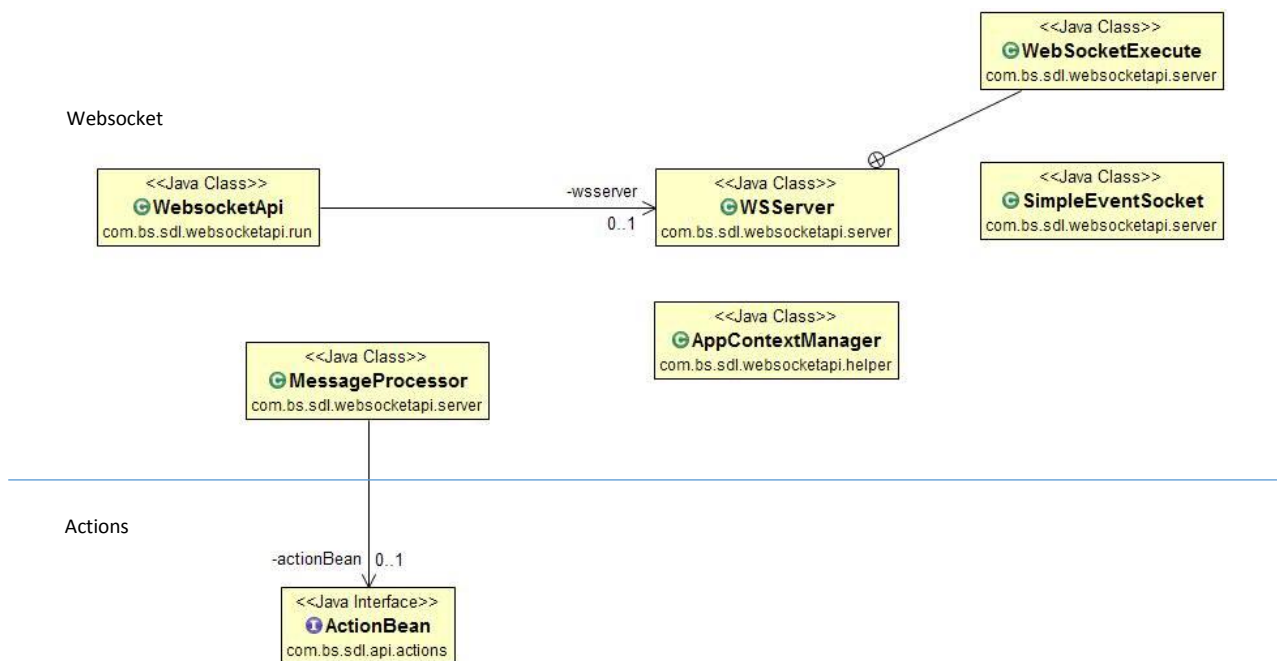
En primer lloc, parlarem de l'arquitectura del SDL (Serveis Dispositius Locals, d'encapsular les peticions a dispositius hardware del caixer) per entendre com l'emulador encaixa i com s'ha aprofitat al màxim tot el que ja s'ha implementat, per després aprofundir en l'emulador i, tot seguit, en els tests i en el llenguatge de programació que s'utilitza per elaborar-los.

10.1.1. Arquitectura del SDL

El SDL segueix una arquitectura on cada capa té la seva interfície: això permet disposar d'un component on les seves capes no depenen les unes de les altres. Aquesta arquitectura segueix un principi de segmentació per interfícies (Interface Segmentation Principle o ISP[7]) i es pot esquematitzar com si fos una torre on cada capa va una sobre l'altra. Cada capa es governa a si mateixa i es comuniquen mitjançant les interfícies.

Aquestes capes, a més a més, divideixen lògicament funcionalitats. Tot seguit es defineix la funcionalitat que li correspon a cada una:

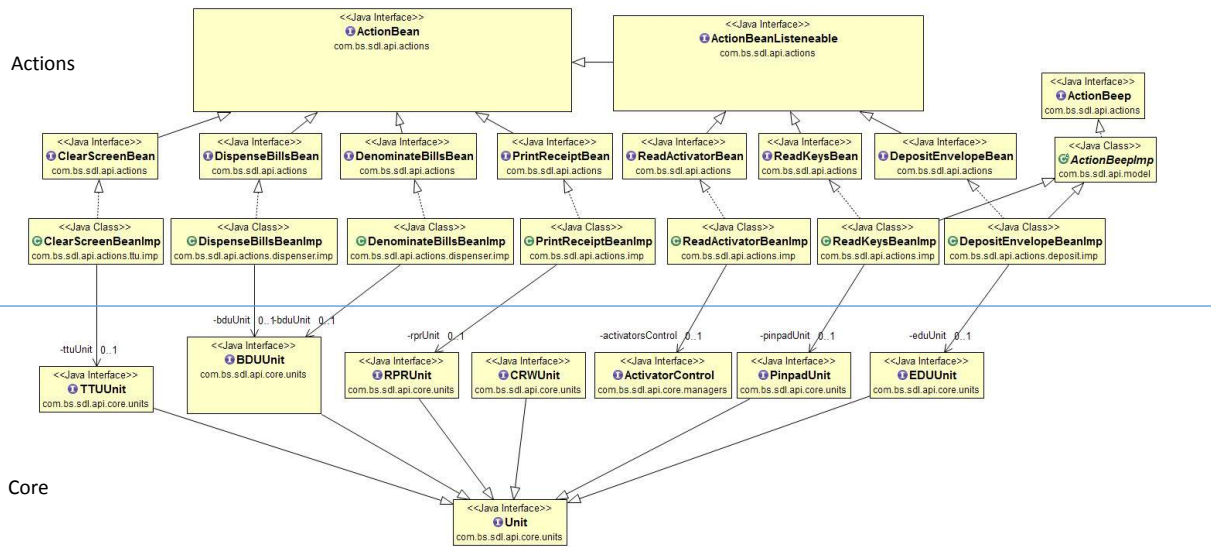
- **WebSocket:** Mitjançant una interfície *websocket*, les vistes, generades al servidor o provinents de l'aplicació offline, són capaces d'interactuar amb un catàleg d'accions que resolen els accessos als dispositius. Tota la interacció amb dispositius es resol de forma local.



Il·lustració 17: Diagrama UML de la capa Websocket i la interfície de la capa Actions

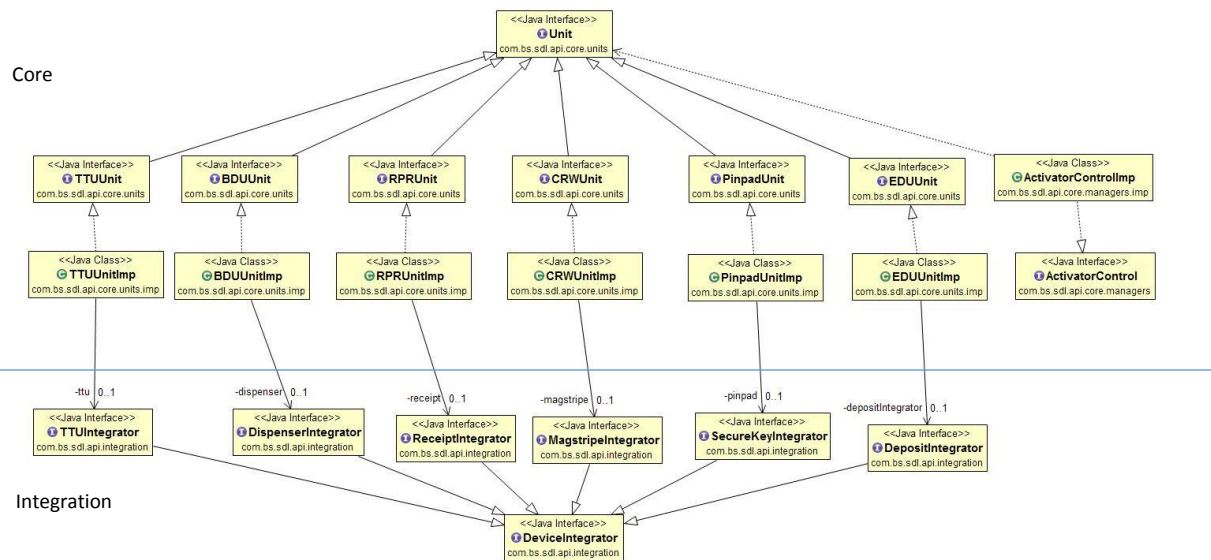
NOTA: La classe de *WSServer* fa ús de *threads* per instanciar la classe *SimpleEventSocket* i aquesta fa el mateix per instanciar la classe *MessageProcessor*. És aquest el motiu per el qual no apareix la relació entre aquestes classes en al diagrama.

- Actions: Un catàleg d'accions sobre els dispositius i components del CORE.



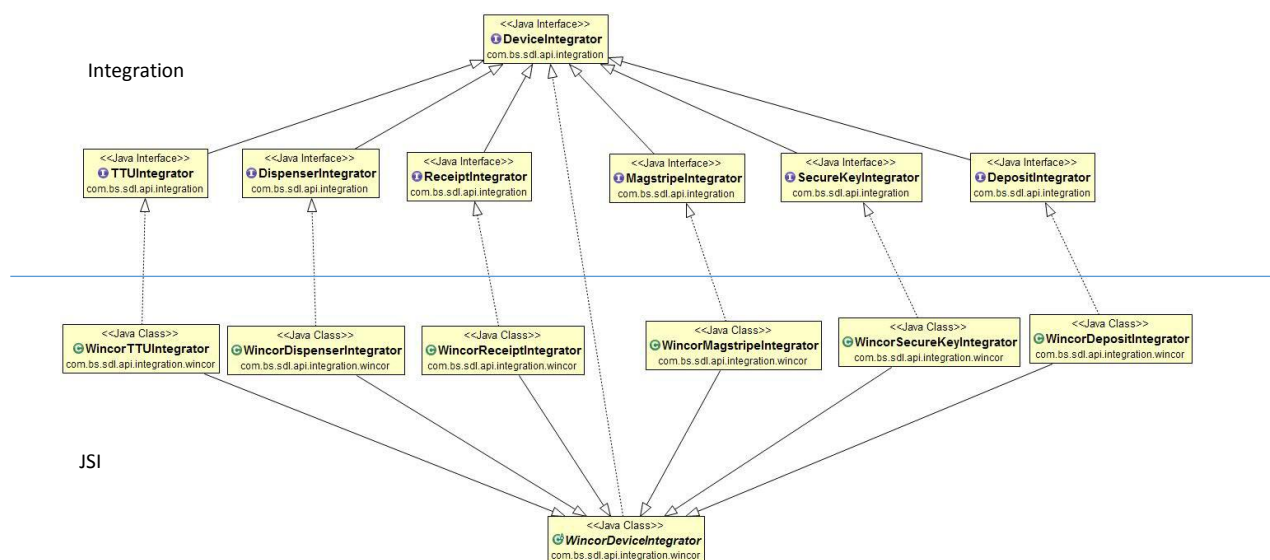
Il·lustració 18: Diagrama UML de la capa Actions i les interfícies de la capa del Core

- Core: Aquesta capa del SDL comprèn aquells components que resolen les interaccions amb els dispositius i la resta de components implicats en les accions exportades per la capa superior.



Il·lustració 19: Diagrama UML de la capa Core i les interfícies de la capa de Integration

- Integration: Ofereix un catàleg d'operacions que encapsulen completament l'accés al *middleware* utilitzat a nivell de dades. Gràcies a l'ús d'interfícies podem disposar de més d'una solució per integrar l'SDL (MDCS o JSI).



Il·lustració 20: Diagrama UML de la capa Integration i les classes de la capa de JSI

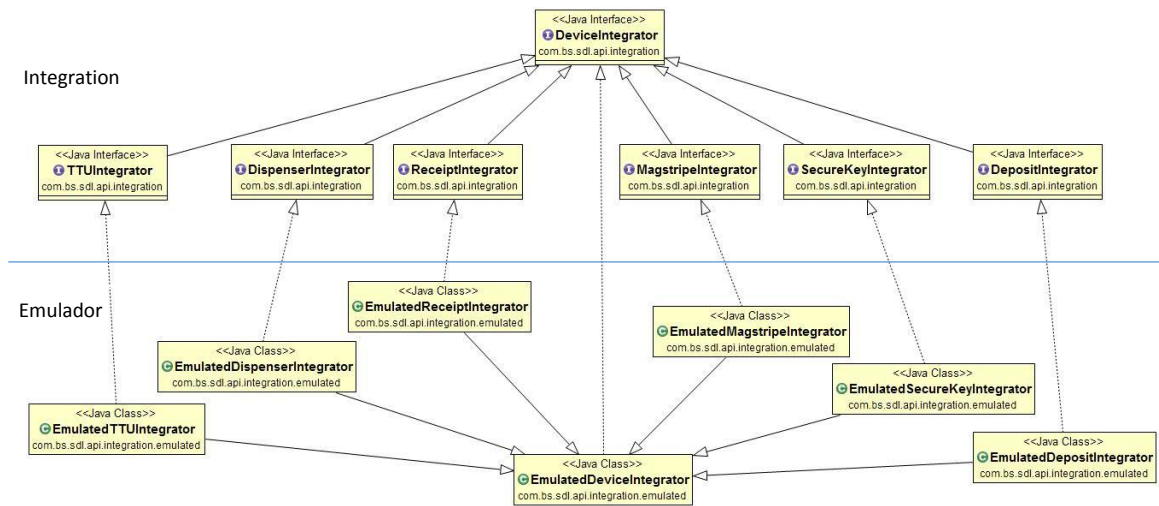
NOTA: Es mostra només la integració amb JSI però la integració amb MDCS vindria a ser un diagrama molt similar.

10.1.2. Arquitectura de l'emulador

Recordem que la funcionalitat principal de l'emulador és resoldre l'accés a dispositius físics, en aquest cas creant una emulació.

Com s'ha vist en l'apartat anterior, les interfícies de la capa *Integration* són implementades per aquelles classes amb la lògica necessària per accedir a dispositius.

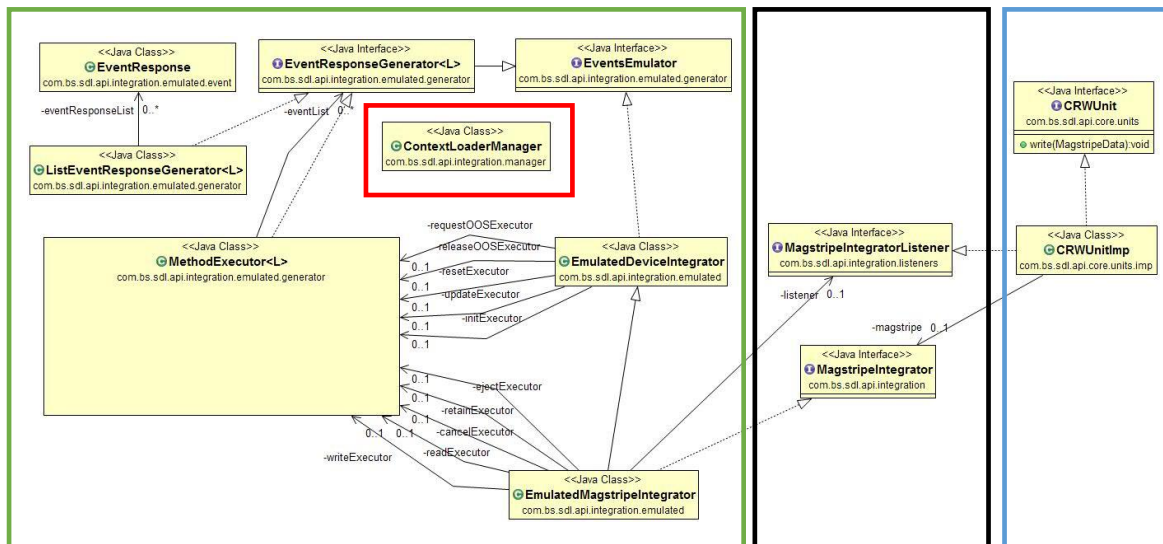
Gràcies a que s'ha seguit un patró ISP, l'emulador ofereix una implementació alternativa de la capa *Integration* amb una lògica concreta. D'aquesta forma podem reaprofitar la resta de capes ja implementades del SDL. El següent diagrama mostra la capa d'integració i les classes de l'emulador que implementen aquestes interfícies.



Il·lustració 21: Diagrama UML de la capa Integration i les classes de l'emulador

10.1.3. Funcionament de l'emulador

Per ajudar a explicar el funcionament de l'emulador usarem el diagrama que hi ha a continuació: correspon a una fracció de l'emulador la qual emula el dispositiu de lectura de targetes. L'explicació es podrà aplicar a la resta de classes de l'emulador.



Il·lustració 22: Diagrama del funcionament de l'emulador

1. Requadre verd: es tracta de l'emulador en si. Podem veure les classes *EmulatedMagstripeIntegrator* i *EmulatedDeviceIntegrator* que són respectivament la classe amb la lògica per poder emular el dispositiu de targetes i la classe pare de totes les classes emuladores. La resta de classes, comunes a per totes les classes emuladores, s'encarreguen de simular la comunicació.

La classe *EmulatedMagstripeIntegrator* emula que una nova targeta ha estat inserida. Quan ha de comunicar-ho a la capa del *core* és quan la resta de classes comunes actuen: mitjançant un *listener* (cada unitat en té un) que comunica la informació necessària generada per l'emulador a la unitat, en aquest cas la *CRWUnitImp* (implementació de la unitat de targetes).

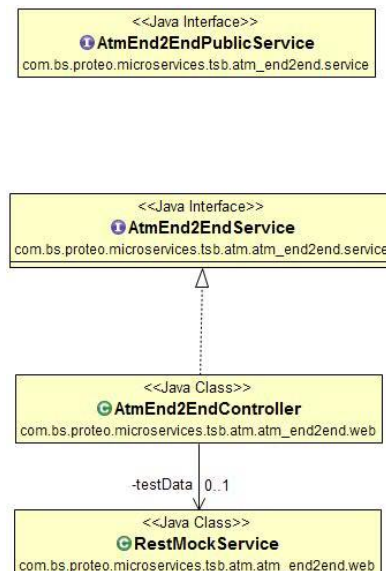
- 1.1. Requadre vermell: mostra la classe que resol la comunicació entre els test i l'emulador (aquesta és la classe principal que resol aquesta comunicació però apareix relacionada amb cap altre classe perquè les classes encarregades de traslladar la informació als emuladors penegen d'aquesta). Aquesta classe és un *endpoint* que els test utilitzen per enviar peticions d'emulació representat com "COM 01" a la il·lustració 16.
2. Requadre negre: interfície de la capa d'integració i el *listener*. Cada classe de l'emulador té la seva interfície com s'ha vist anteriorment (il·lustració 21) i a més a més tenen un *listener*, que també realitza la funció d'interfície on s'indica quins mètodes es podran cridar des de fora de la unitat. Amb altres paraules: les classes que implementin la capa *Integration* usaran aquests mètodes per proporcionar informació a les unitats.
3. Requadre blau: cada unitat implementa la seva interfície i seu *listener*. D'aquesta forma podem tenir més d'una implementació de la mateixa unitat amb diferents *listeners*.

10.2. Mòdul de respostes

Tal com es mostra a la il·lustració 16, el mòdul de respostes és el component que serveix respostes a l'entitat Proteo4 SSMM (formada per Proteo4 ATM Channel i Proteo4 API Services) evitant haver de cridar als *backends* de l'entitat financera.

10.2.1. Arquitectura del mòdul de respostes

Aquest mòdul es compon de dues interfícies: la primera *AtmEnd2EndPublicService* és la interfície que utilitzarà Proteo4 API Services en mode test per obtenir respostes. I la segona és la *AtmEnd2EndService*, que és la interfície que els test utilitzen per demanar la càrrega d'una resposta en concret (etiquetat com "COM 02" de la il·lustració 16). I per últim la classe *RestMockService* és el catàleg de possibles respostes a carregar.



Il·lustració 23: Diagrama mòdul de respostes

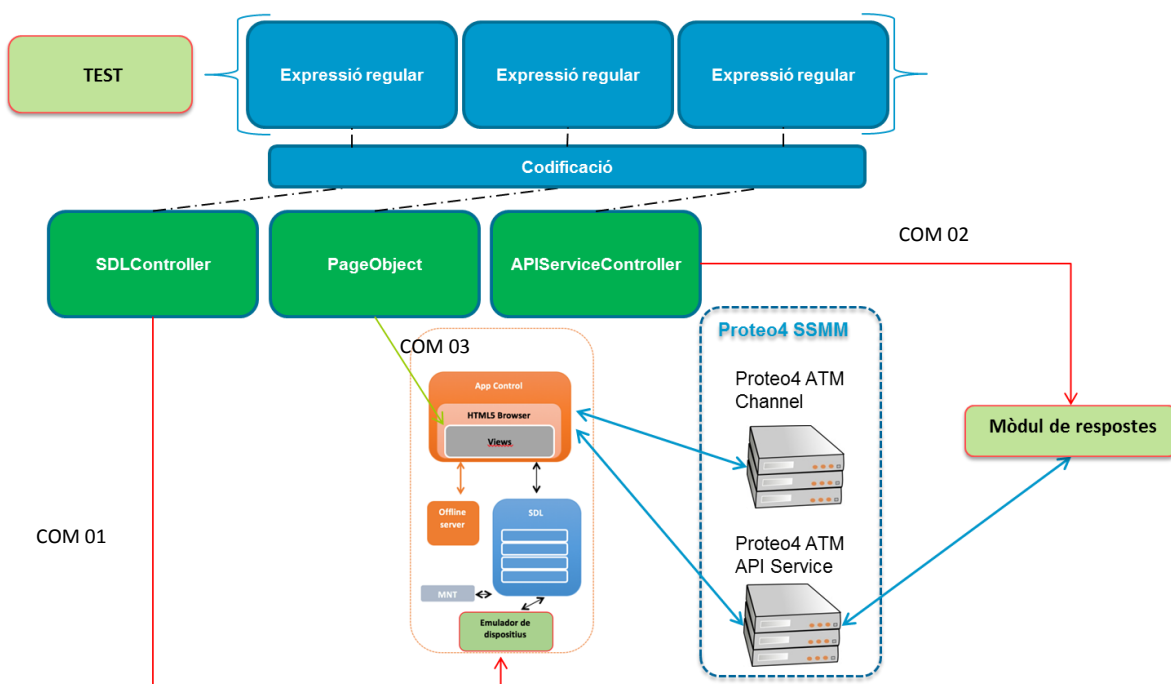
10.2.2. Funcionament del mòdul de respostes

El funcionament d'aquest mòdul consisteix en atendre les peticions que fan els tests (a la interfície *AtmEnd2EndService*): carrega la resposta demanada a la classe *RestMockService* que posteriorment serà retornada al Proteo4 API Services quan demani una resposta a la interfície *AtmEnd2EndPublicService*.

10.3. Arquitectura de tests

El següent diagrama és el mateix que el de l'inici d'aquest capítol (il·lustració 16) però en aquest cas els tests estan desglossats en els components que el conformen.

En els apartats que segueixen, explicaré quina és la funció de cada d'aquest components i com funciona.



Il·lustració 24: Diagrama de l'arquitectura dels tests

10.3.1. Escriptura i automatització dels tests

Un dels objectius d'aquest projecte és poder escriure i automatitzar els tests. Avui en dia existeixen eines de desenvolupament que proporcionen aquestes prestacions. Així doncs, per solucionar aquesta part del problema i evitar reinventar la roda es va decidir utilitzar un software de desenvolupament anomenat Cucumber[8].

Cucumber aporta alguns aspectes clau poder escriure i automatitzar els tests, com veurem a continuació.

En primer lloc, Cucumber treballa amb Gherkin, el llenguatge que utilitza per definir tests. Gherkin és un llenguatge *line-oriented*, és a dir cada línia és una instrucció sempre i quan la línia no estigui en blanc o no sigui un comentari. A més a més, disposem de l'avantatge que programar amb Gherkin és pràcticament com escriure en llenguatge natural, la qual cosa facilita la comprensió per part d'experts funcionals.

Si ens situem a la il·lustració 24, la programació en Cucumber serien els requadres blaus de la part superior del diagrama, on es pot llegir “Expressió regular” o “Codificació”. En els que es pot llegir “Expressió regular” tenen un aspecte com la següent il·lustració: un fitxer amb diferents escenaris on cada escenari es compon d’expressions regulars.

```
1  Feature: TEST_000
2
3  Scenario: TSB debit
4
5      Given an ATM in service in a "UK" office
6          And the ATM shows the "attract" screen
7      Then I insert a "TSB debit" card
8          And my card is "OK"
9      Then the ATM shows the "PIN entry" screen
10         And I press the key "1"
11         And I press the key "2"
12         And I press the key "3"
13         And I press the key "4"
14
15
16  Scenario: TSB credit
17
18  Given an ATM in service in a "UK" office
19      And the ATM shows the "attract" screen
20  Then I insert a "TSB credit" card
21  .
22  .
23  .
24  .
```

Il·lustració 25: Exemple d'una feature

Els fitxers que, com l’exemple anterior, contenen escenaris els anomenarem *feature*. Cal destacar que les paraules entre cometes dins de les expressions (per exemple “TSB dèbit” o “PIN entry”) són expressions regulars parametritzades: això ens permet reutilitzar expressions regulars i poder fer codificacions d’aquestes expressions genèriques i, segons el paràmetre, també comprovacions específiques.

Pel que fa al segon requadre blau, mencionat anteriorment, on es pot llegir “Codificació” aquest és el fitxer on es codifica el que farà cada expressió regular escrita en llenguatge natural.

La il·lustració 26 mostra com seria l’estructura del fitxer de codificacions. En concret l’esquelet mostrat seria per les expressions definides en la *feature* de la il·lustració 25).

```

this.Given(/^an ATM in service in a "(.*)" office$/, function (officeLocation, callback) {
    <Codificació de l'expressió regular de la línia 5 i 18>
});

this.Given(/^the ATM shows the "(.*)" screen$/, function (screen, callback) {
    <Codificació de l'expressió regular de la línia 6 i 19>
});

this.Then(/^I insert a "(.*)" card$/, function (cardType, callback) {
    <Codificació de l'expressió regular de la línia 7 i 20>
});

this.Then(/^my card is "(.*)"$/, function (cardState, callback) {
    <Codificació de l'expressió regular de la línia 8>
});

this.Given(/^the ATM shows the "(.*)" screen$/, function (screen, callback) {
    <Codificació de l'expressió de la línia 9>
});

this.Then(/^I press the key "(.*)"$/, function (key, callback) {
    <Codificació de l'expressió regular de la línia 10, 11, 12, 13>
});

```

Il·lustració 26: Exemple de codificació d'expressions regulars

En resum, Cucumber executarà els fitxers *.feature* indicats, que són conjunts d'expressions regulars codificades. Un cop s'hagi executat satisfactòriament la codificació d'una expressió regular es passarà a la següent. I si totes les expressions d'un escenari s'executen correctament el test ha passat amb èxit.

10.3.2. Tipus de codificació de les expressions regulars

Un cop sabem com s'escriu un test a partir d'expressions regulars, cal que disposi d'una codificació.

Observant de nou la il·lustració 24, ens centrarem en els requadres verds: cada un d'ells (SDLController, PageObject i APIServiceController) representa un tipus de codificació d'expressions regulars.

- **SDLController:** És un mòdul on hi ha la codificació per totes les expressions regulars que interactuen amb l'emulador. Es fan peticions al *endpoint* de l'emulador, etiqueta "COM 01" de la il·lustració 24, per disposar del patró de càrrega desitjat en cada moment.

A continuació es pot veure un exemple de com seria una petició del test a l'emulador per simular que s'ha premut la tecla 1 del teclat, i la resposta satisfactòria a la petició.

Petició:

```
{"action": "SDL_CLM_CONTEXT",  
  "operation": "LOAD",  
  "emulator": "securekey",  
  "bean": "securekey-readkey",  
  "eventExecutor": "read",  
  "params": {"key": "1"}}
```

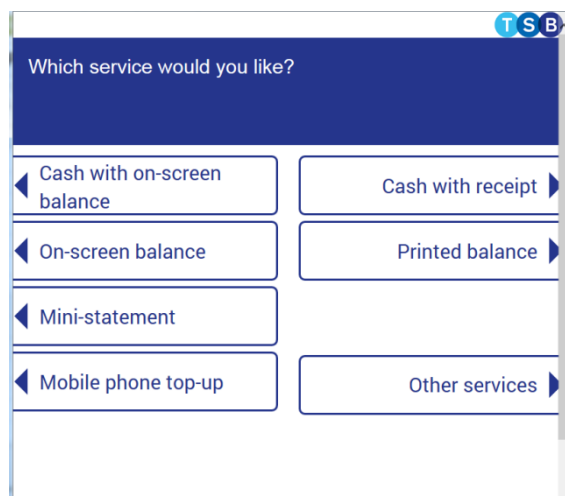
Resposta:

```
{"action": "SDL_CLM_CONTEXT-LOAD",  
  "result": "OK SDL_E2E",  
  "loadedEvents": "EventResponse [delay=100, eventName=keyRead,  
    eventData=[ResultDetail [resultCode=RSLT_SUCCESS,  
      specificCode=0, detail=detail] KeyCode= KEY_0]]"}
```

- PageObject: Aquest mòdul engloba totes les expressions regulars de comprovació de components visuals que podrà veure l'usuari final.

A la il·lustració 24 podem veure que hi ha una etiqueta "COM 03" senyalant el *browser*, que mostra el funcionament d'aquest mòdul. Consisteix en recuperar cada una de les vistes que es mostren en el navegador (fitxers HTML) i comprovar que inclou tots els elements requerits: per exemple si un test passés pel següent menú (il·lustració 27) el que comprovaria seria:

- la disposició dels botons i la comprovació que tots els que han de ser visibles ho són
- el títol apareix i l'idioma del títol és el que toca
- el logo hi és i és visible



Il·lustració 27: Menú

- **APIServiceController:** És un mòdul molt semblant al **SDLController** però aquest codifica la comunicació amb el mòdul de respostes (etiqueta “COM 02” en la il·lustració 24) de l’**APIService**. El test comunica en cada moment quina és la resposta que desitja carregar perquè **Proteo4 ATM APIService** la consumeixi.

A continuació podem veure la petició que es faria al mòdul de respostes si volem carregar una resposta de PIN erroni que **Proteo4 ATM APIService** consumirà en un futur i la resposta satisfactòria a la petició.

Petició:

```
{
  "headers": {
    "HEADER.TRACKING_ID": "DCRL0100Scc8f3c1b-558b-499d",
    "HEADER.SERVICE_LANGUAGE": "ENG",
    "HEADER.SERVICE_COUNTRY": "ENG",
    "HEADER.SERVICE_CHANNEL": "atm",
    "HEADER.SERVICE_PROCEDEENCE": "atmid",
    "HEADER.APPLICATIONID": "atm",
    "ATM.HEADER.CONFIG.ATMID": "15",
    "ATM.HEADER.CONFIG.MODEL": "MODEL",
    "ATM.HEADER.LANGUAGE": "EN",
    "ATM.HEADER.CONFIG.OFFICE": "ENGLAND_0001"},
  "body": {
    "testId": "atmbalance-v0/atmbalance/wrongpintwice",
    "values": {
      "retainedCard": "false"
    }
  }
}
```

Resposta:

```
{
  "headers":{
    "HEADER.STATUS": "0",
    "HEADER.TRACKING_ID": "DCRL0100Scc8f3c1b-558b-499d" },
  "body": {
    "accepted": "true"
  }
}
```

11. Proves

En aquest apartat explicaré quines són les proves que s'han realitzat per validar el funcionament de l'emulador i el mòdul de respostes. També descriu com es realitzen les proves *end-to-end*.

11.1. Proves de l'emulador i del mòdul de càrrega

Els tests unitaris de l'emulador s'han definit per cada dispositiu que s'ha emulat. Això correspon a un test unitari per cada classe de la capa Emulador (il·lustració 21) on aquest test prova totes les casuístiques i casos extrems del codi escrit.

Pel que fa al mòdul de respostes, els tests que s'han realitzat consisteixen en provar que es pot fer una petició externament i que el que retorna és efectivament el que s'havia demanat.

En tots dos casos, per passar els test unitaris s'utilitza el *framework* de TestNG mencionat en l'apartat anterior, i es passen cada vegada que volem compilar una nova versió perquè hi ha canvis. Aquest tests són els primers a detectar errors en el codi.

11.2. Proves *end-to-end*

Un cop tenim els test unitaris de l'emulador i el mòdul de respostes, només queda poder passar tests.

El component test, que recordarem que està format per les expressions regulars i la seva codificació, no té tests unitaris. Que el propi test passi amb èxit és la validació de que les expressions regulars són correctes, que la codificació d'aquestes està ben escrita i que els components emulador i mòdul de respostes han funcionat, tot això a part de la garantia que teníem que aquest dos últims ja passaven els seus tests unitaris.

Un cop sabem quins test passem per cada component, només queda saber com s'acaben realitzant les proves d'una funcionalitat completa del sistema o *end-to-end*. Per fer això disposem d'una màquina dedicada a la integració contínua on hi ha un servidor *Jenkins*. En aquest servidor hi ha un *job*, o *script* d'execució, que s'encarrega d'automatitzar el desplegament de l'arquitectura de l'*E2E-framework*.

En resum, gràcies a tenir una arquitectura que passa tests de forma automàtica --un dels objectius del projecte-- i a la màquina d'integració, es pot programar quins tests es volen passar i quan es vol que passin. I un cop acabada cada l'execució podran observar-se informes molt visuals sobre com ha anat l'execució.

Com a exemple, tot seguit es pot veure l'informe de l'execució de les proves *end-to-end*, detallant de tots els *steps* que s'han executat, quants han passat, han fallat o s'han saltat o no estaven definits.



Il·lustració 28: Informe post execució test end-to-end: expressions regulars

O per exemple aquest altre, una taula amb tots els tests executats i un recompte primer a nivell dels escenaris i després a nivell d'expressions regulars i finalment el temps invertit en cada un dels test i l'estat (si ha passat o ha fallat).

Feature	Scenarios			Steps							Duration	Status
	Total	Passed	Failed	Total	Passed	Failed	Skipped	Pending	Undefined	Missing		
JNIT	1	1	0	1	1	0	0	0	0	0	02s 393 ms	Passed
TEST_OSB_0009	1	0	1	29	4	1	24	0	0	0	12s 192 ms	Failed
TEST_OSB_0010	1	0	1	37	4	1	32	0	0	0	05s 267 ms	Failed
TEST_OSB_0011	1	0	1	35	4	1	30	0	0	0	09s 416 ms	Failed
TEST_OSB_0012	1	0	1	29	4	1	24	0	0	0	05s 148 ms	Failed

Il·lustració 29: Informe post execució test end-to-end taula de tests

A més a més aquesta taula permet veure el detall de cada test. Mostrant *step* per *step* com ha anat l'execució. La següent il·lustració ho exemplifica i es pot observar els *steps* que ha passat i els que han fallat i els *steps* que després de la fallada s'han saltat.

Feature	Scenarios			Steps							Duration	Status
	Total	Passed	Failed	Total	Passed	Failed	Skipped	Pending	Undefined	Missing		
TEST_OSB_0009	1	0	1	29	4	1	24	0	0	0	12s 192 ms	Failed
Feature TEST_OSB_0009												
Scenario OSB 0009: TSBd Error PIN 1												
Given an ATM in service in a "UK" office											000 ms	
And the ATM shows the "attract" screen											02s 904 ms	
Then I insert a "TSB debit" card											02s 166 ms	
And my card is "OK"											02s 118 ms	
Then the ATM shows the "PIN entry" screen											05s 001 ms	
Error message												
And I press the key "1"											000 ms	
And I press the key "2"											000 ms	
And I press the key "3"											000 ms	
And I press the key "3"											000 ms	

Il·lustració 30: Informe post execució test end-to-end detall d'un test

12. Tecnologies i eines emprades

Tot seguit esmentaré les tecnologies i eines usades en aquest projecte. Estan dividides en quatre taules:

Tecnologies emprades per al desenvolupament	
1. Jetty	Es un motor per <i>Servlets</i> i suporta HTTP i <i>WebSocket</i> entre altres
2. Spring Framework	<i>Framework</i> de codi obert per Java
3. Maven	Software de gestió de dependències de projectes
4. Java	Llenguatge de programació de propòsit general i orientat a objectes
5. JavaScript	Llenguatge de programació de <i>scripting</i> basat en prototipus
5. JSON	Estàndard de comunicació entre objectes que utilitza text llegible per humans
6. HTML 5	Llenguatge de programació <i>markup</i> (o marcat) utilitzat per crear pàgines web
7. Grunt	Tecnologia que permet automatitzar tasques de compilació, d'empaquetat o de <i>testing</i>
8. Protractor	<i>Framework</i> per execució de tests <i>end-to-end</i> per aplicacions en Angular JS
9. Selenium	Tecnologia que permet automatitzar navegadors
10. Cucumber	Tecnologia per executar tests funcionals automàticament
11. TestNG	<i>Framework</i> per Java per passar tests

Taula 4: Tecnologies i eines emprades I (desenvolupament)

Eines emprades per la gestió de codi	
1. Git	Sistema de control de versions de codi
2. Bitbucket	Sistema de control de versions de codi
3. Eclipse	Entorn integrat de desenvolupament de codi obert pensat per programar principalment en Java
4. SublimeText	Editor de codi que suporta multitud de llenguatges de programació

Taula 5: Tecnologies i eines emprades II (gestió de codi)

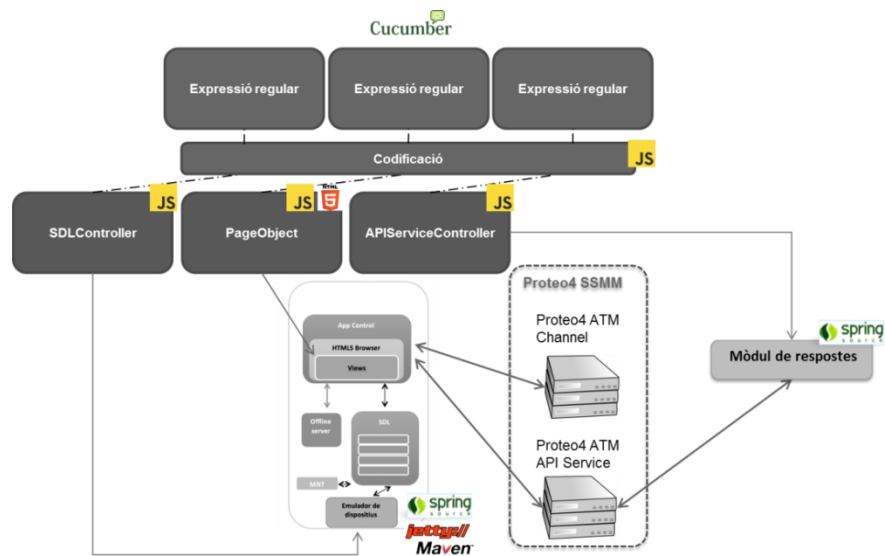
Eines emprades per la gestió de documentació i equip	
1. Confluence	Sistema de col·laboració i difusió d'informació en l'entorn del projecte
2. Google Docs	Processador de textos i fulls de càlcul en línia
3. Outlook	Client de correu electrònic
4. Gmail	Client de correu electrònic
5. JIRA	Software de gestió de tasques

Taula 6: Tecnologies i eines emprades III (gestió documentació i equip)

Eines emprades per a la realització de la memòria	
1. MS Word	Processador de textos
2. MS Excel	Fulls de càlcul
3. MS PowerPoint	Disseny de presentacions
4. Draw io	Diagrames en general
5. ObjectAid	Diagrames UML

Taula 7: Tecnologies i eines emprades IV (realització de la memòria)

Per últim es mostra una il·lustració de l'E2E-framework, on s'ha sobreimprès les tecnologies amb les que s'ha desenvolupat cadascun dels components.



Il·lustració 31: Tecnologies emprades per component

13. Gestió econòmica

13.1. Identificació i estimació dels costos

A continuació s'identifiquen els costos del projecte. Dividits en quatre taules: recursos humans, recursos hardware, recursos software i despeses generals. També es te present el cost dels possibles imprevistos i una reserva de contingència. Com s'ha dit a l'apartat "8. Planificació", es un projecte de 160 dies amb una dedicació de 4h diàries és a dir 640 hores de projecte.

13.1.1. Recursos humans

Remuneració corresponent a cada rol que participa al projecte.

Rol	Hores	Remuneració per hora	Cost en €
Cap de projecte	250	40,00	10.000,00
Analista	80	15,00	1.200,00
Programador	210	8,50	1.785,00
Tester	100	11,67	1.167,00
Total estimat	640		14.152,0

Taula 8: Despeses recursos humans

13.1.2. Recursos hardware

Es calcula que els recursos hardware tindran una vida útil de 4 anys. On cada any són 249 dies hàbils, amb un ús de 4 hores diàries.

Hardware	Preu en €	Hores	Amortització en €	Cost en €
Portàtil Dell	750	600,00	0,188	141,190
Teclat Microsoft	25,5	500,00	0,006	0,163
Ratolí Microsoft	10	500,00	0,003	0,025
Monitor Dell	200	500,00	0,050	10,040
Total estimat				151,42

Taula 9: Despeses recursos hardware

Amortització en € = Preu en € / (4 anys * (249 dies * 4 hores diàries)).

13.1.3. Recursos software

Es calcula que els recursos hardware tindran una vida útil de 3 anys. On cada any són 249 dies hàbils.

Software	Preu en €	Hores	Amortització en €	Cost en €
Windows 7	250	500,00	0,084	20,92
Elipse Mars	0	-	0,000	0,00
Microsoft Office Suite	500	180,00	0,167	83,67
Git, Bitbucket	0	-	0,000	0,00
Google Drive	0	-	0,000	0,00
Outlook, Gmail	0	-	0,000	0,00
Grunt	0	-	0,000	0,00
Sublime Text	0	-	0,000	0,00
Total estimat				104,59

Taula 10: Despeses recursos software

Amortització en € = Preu en € / (3 anys * (249 dies * 4 hores diàries)).

13.1.4. Despeses generals

Totes aquelles despeses relacionades amb l'ús d'oficina on es treballa.

Despesa	Preu al mes		Cost en €
Llum i aigua	50	Fixat per l'empresa.	400,0
Fibra òptica	100	Fixat per l'empresa.	800,0
Material Oficina	12	Fixat per l'empresa.	96,0
Total estimat			1.296,00

Taula 11: Despeses generals

13.1.5. Imprevistos

En el cas de que el projecte pateixi desviacions en les estimacions realitzades hi ha una reserva en el pressupost per poder reaccionar. Assumim que això pot succeir en un risc del 35% de manera que això afectaria a la planificació del projecte. Si fos necessari, les dues últimes setmanes es podria ampliar la de dedicació de 4 a 6 hores és l'equivalent a 5 dies amb una dedicació de 4h.

La fórmula per a aquest càlcul seria:

$$\text{Despeses imprevistos} = 0,35 ((\text{despeses recursos humans} + \text{despeses recursos hardware} + \text{despeses recursos software} + \text{despeses generals}) / (\text{hores totals del projecte} * 4 \text{ hores diàries})) * 5 \text{ dies} = 6,08\text{€/dia} * 5 \text{ dies} = 30,37\text{€}.$$

13.1.6. Contingència

Com a precaució s'estableix una mesura de contingència del 7% sobre el pressupost total del projecte.

13.1.7. Cost total

En resum, les costos totals del projecte es resumeixen en la taula següent:

Despesa recursos humans	14.152,0
Despesa recursos hardware	151,42
Despesa recursos software	104,59
Despesa general	1.296,0
Imprevistos	30,37
Cost parcial	15.734,38
Contingència 7%	1.101,4
Cost total	16.836,0

Taula 12: Cost total

13.2. Seguiment i control del projecte

Com en tot projecte, cal dur a terme un seguiment acurat de la seva execució, tant pel que fa a l'execució com als costos. En aquest cas, la desviació els costos només pot venir per un allargament del termini i/o per requerir una major dedicació en hores deguda a canvis de requeriments.

Per tant, es farà un seguiment de com evolucionen els desenvolupaments previstos a través de les tasques i si s'assoleixen les fites. I gràcies a la metodologia iterativa que s'ha seguit (vegeu l'apartat "7. Metodologia del treball") es pot garantir una adequada supervisió i control de la dedicació d'hores al projecte.

14. Sostenibilitat i compromís social

Els indis Iroquesos, originaris de la zona dels Grans Llacs, ja contemplaven el Principi de les set generacions[9], que establí que els caps havien de considerar sempre els efectes de les seves accions sobre els seus descendents fins a la setena generació propera.

Avui en dia analitzem la repercussió de les nostres actuacions des del concepte sostenibilitat (Informe Brundtland, 1987[10]), entesa com la capacitat de “proveir el millor possible avui sense posar en perill els recursos de demà”.

Aquesta previsió ha d'estar present en tot projecte que emprenem avui en dia, i cal mesurar-la des de les seves tres dimensions bàsiques: sostenibilitat econòmica, sostenibilitat social i sostenibilitat mediambiental.

En el cas del projecte que estem exposant, cal dir que es tracta d'una petita part d'un projecte de molt més abats, per la qual cosa la seva repercussió directa és força limitada. Tot i això, hem pogut identificar igualment diversos aspectes de la seva sostenibilitat

14.1. Sostenibilitat econòmica

Els recursos utilitzats són adequats i proporcionats a la consecució de l'objectiu i la dimensió del projecte.

En tractar-se d'un projecte que segueix una metodologia àgil, que afavoreix la revisió constant de la seva execució, evitem generar artefactes d'un sol ús i en centrem aportar valor al projecte amb cada tasca que es completa.

Podem considerar que el projecte és sostenible econòmicament ja que s'han avaluat els recursos necessaris i descrit protocols que assegurin el control i ajust, en cas de necessitat, durant tota la vida del projecte.

També cal tenir en compte que, com més provat i testejat estigui un programari adreçat a l'usuari final major satisfacció obtindrà i per tant oferirà una major durabilitat en el temps i menys necessitats de canvi o de substitució, amb el consegüent estalvi de recursos.

Finalment, cal tenir present que, pel fet de tenir codificats els tests, s'estalvia temps en el desenvolupament i s'eviten els errors o omissions que es podrien produir realitzant-los manualment.

14.2. Sostenibilitat social

Aquest projecte permetrà a tot un equip, el de *ATM for TSB*, recolzar-se en una plataforma d'emulació, evitant necessitar un dispositiu físic que hauria estat més costós i menys eficient.

L'objectiu del nostre projecte, és poder realitzar el màxim de proves amb rapidesa i agilitat, i poder-les repetir tantes vegades com calgui, la qual cosa redundarà, necessàriament, en una major facilitat i seguretat per a l'usuari final.

A més a més, el nostre desenvolupament s'orienta a persones de cultures diferents, de tal manera que sigui reutilitzable en contextos culturals distints, sense que calgui fer-hi modificacions.

Podem considerar, doncs, que el projecte és sostenible socialment pel fet que impacta de forma directe a tothom qui està implicat des de el desenvolupador fins al client final, i cal tenir-ho present, per complir els drets legalment i amb la responsabilitat ètica i professional.

14.3. Sostenibilitat ambiental

El major consum en aquest projecte és elèctric. Tot els recursos són virtuals per tant no hi ha un gran consum de matèries primeres.

Podem considerar que el projecte és sostenible ambientalment els plans de vida útil dels ordinadors utilitzats és raonable i la majoria de tràmits o artefactes són digitals. I si un d'aquest artefactes existeix, per exemple en el projecte de *ATM for TSB*, el reutilitzem. L'ús responsable de tots els recursos és promogut en l'entorn de treball, principalment aquells recursos provinents de matèries primeres.

15. Aspectes legals

A continuació tractarem els consideracions legals d'aquest projecte. La següent taula mostra les llicències de les tecnologies usades.

Jetty	Apache Licence 2.0
Spring Framework	Apache Licence 2.0
Maven	Apache Licence 2.0
Java	GNU General Public Licence
JavaScript	MIT License
Grunt	MIT License
Protractor	MIT License
Selenium	Apache Licence 2.0
Cucumber	MIT License
TestNG	Apache Licence 2.0

A continuació es dona el detall dels drets que tenim sobre el software segons cada una de les tres llicències que s'han identificat:

-Apache Licence 2.0[11]: L'usuari te dret a: executar el software per qualsevol propòsit, distribuir-lo , modificar-lo i distribuir el software modificat.

-GNU General Public Licence[12]: L'usuari te dret a: executar, estudiar, copiar, compartir i modificar el software.

-MIT Licence[13]: L'usuari te dret a: executar, copiar, modificar, publicar, distribuir i vendre còpies del software

Aquesta projecte a més a més té un marc legal imposat per el client a part de les llicències mencionades. Aquest imposa unes condicions i també guies d'arquitectura (per exemple la de Proteo4). Una d'aquestes condicions és que el codi d'aquest projecte és propietat seva i no el podem distribuir.

16. Assoliment de les competències tècniques

En aquest apartat revisaré les competències tècniques que en el moment de la inscripció d'aquest projecte vaig dir que es treballarien. Aniré competència a competència justificant l'assoliment.

CES1.1: Desenvolupar, mantenir i avaluar sistemes i serveis software complexos i/o crítics. [Bastant]

- En aquest projecte s'ha assolit aquest competència, al avaluar un sistema complex, el dissenyat per l'equip d'*ATM for TSB*, i tot seguit desenvolupar el *E2E-framework* i mantenir-lo.

CES1.2: Donar solució a problemes d'integració en funció de les estratègies, dels estàndards i de les tecnologies disponibles. [En profunditat]

- En aquest projecte s'ha assolit aquest competència en profunditat, al fer la tria i disseny de com construir els test i quines tecnologies utilitzar, com per exemple, el mòdul de respostes permet, que fer proves *end-to-end* i evita les crides als *backends*.

O també amb la integració de l'emulador amb el SDL, reaprofitant el màxim de codi possible al utilitzar el principi de segmentació per interfícies (Interface Segmentation Principle o ISP).

CES1.3: Identificar, avaluar i gestionar els riscos potencials associats a la construcció de software que es poguessin presentar. [Una mica]

- En aquest projecte s'ha assolit aquest competència, al fer estimacions i planificacions de les iteracions. i tenint en compte els riscos identificats. Aquestes prediccions s'han ajustat o el seu possible impacte s'ha tingut en compte en format de contingència.

CES2.1: Definir i gestionar els requisits d'un sistema software. [Bastant]

- En aquest projecte s'ha assolit aquest competència, al definir un software que interactua amb altres sistemes i amb usuaris. Ha sigut necessari identificar els requisits de l'*E2E-framework* i després tenir-los presents durant la fase de construcció.

17. Conclusions i treball futur

En el transcurs d'aquest projecte s'ha creat un *framework* per poder validar aplicacions web amb dependències d'interacció humana amb caixers automàtics. La solució aportada cobreix tot l'abast del problema que es plantejava a l'inici. Per tant els objectius proposats s'han assolit.

La planificació temporal que es va realitzar s'ha complert, amb petits ajustos però totes les tasques planificades s'han realitzat. Això implica que s'han implementat satisfactòriament tots els casos d'ús proposats. I a hores d'ara ja s'està treballant en nous casos d'ús.

A més a més d'assolir el *milestone* proposat aquest amb projecte he tingut la oportunitat de viure el dia a dia d'un projecte empresarial. Aprendre i participar en la dinàmica de desenvolupament d'un equip que treballar amb metodologies de desenvolupament àgils. Participant en els esdeveniments, com els *sprint planning* o el *daily scrum*.

Poder formar part d'aquest projecte m'ha aportat molt coneixement en arquitectura i disseny de *software*. Per exemple freqüentment havia de preparar una proposta de disseny o tractar amb nous components i entendre com s'han codificat. També he tingut l'oportunitat d'estudiar components software complexes per poder analitzar les decisions i patrons que m'han ajudat a millorar el meu coneixement d'arquitectura del software.

Simultàniament, al tractar-se d'un projecte per al sector bancari he adquirit coneixement d'on resideix i com es duu a terme aquest negoci, i més específicament la part més relacionada amb infraestructura, gestió i funcionament dels caixers automàtics.

Pel que fa al desenvolupament actual del projecte, tot i que ja hem completat el *milestone*, no hem arribat encara al final del projecte i queda feina i espai per millorar: per exemple no estem emulant encara tots els dispositius dels que disposa un caixer. En un futur caldrà seguir afegint dispositius emulats, patrons, respostes i tests per provar aquests nous dispositius.

En segon lloc hi ha una proposta de crear un component que permeti al *E2E-framework* executar més d'una instància del *framework* al moment, proporcionant la possibilitat de paral·lelitzar i balancejar execucions de tests molt llargues.

I un últim punt en possibles millores és en l'àmbit de l'experiència d'usuari. El *E2E-framework* serveix un propòsit però en aquests moments tota la interacció entre el sistema i l'usuari transcorre en una línia de comandes i no sol ni ser la millor opció ni la més agradable per usuaris no tècnics. Per tant es podria projectar el desenvolupament d'una interfície d'usuari on la interacció, redacció de tests i presentació de resultats fos molt més intuïtiva.

Per concloure, aquest projecte ha complert els objectius i la planificació marcada. Tot i que encara queda treball futur hi ha ganes i emoció per part de l'equip de seguir endavant. I a més a més, a nivell personal, he adquirit molt coneixement en un camp que m'apassiona el de desenvolupament software. Ha estat una experiència molt valuosa i plena de reptes.

Moltes gràcies a tots els qui heu estat presents en aquest projecte per l'oportunitat que he pogut viure.

Glossari

Framework: un conjunt de conceptes, pràctiques i criteris per solucionar un problema en concret i aquestes serveixen per encarar i resoldre problemes d'índole similar.

ATM: acrònim de *Automated Teller Machine* la traducció és caixer automàtic.

Browser: un navegador web.

Proteo 4: guies i model d'arquitectura proposada per el client.

Testing: metodologia de realització de proves o test per confirmar el funcionament del codi.

End-to-end (E2E): indicació de que la funcionalitat del sistema s'utilitza o prova en la seva totalitat.

Stakeholders: tots els actors implicats en un projecte.

Backends: base de dades on s'emmagatzema informació crítica de negoci.

API: una interfície programable d'aplicacions. Ofereix una llibreria de funcions que es poden usar des de un altre software.

Milestone: objectiu o fita.

Emulador: component que imita el comportament d'un altre sistema o dispositiu

E2E-framework: nom del framework desenvolupat

Dispositius: sensor hardware amb un objectiu concret

Magstripe: dispositiu de lectura de targetes

Receipt: dispositiu d'impressió de rebuts

Securekey: dispositiu de tecles

Dispenser: dispositiu de entrega de bitllets

EndPoint: la URL on es publica un servei per que aplicacions client hi pugin accedir.

Core: nucli, classe o classes que conformen la funcionalitat principal.

Scrum: metodologia àgil i iterativa de desenvolupar software.

Listener: instrument de software que serveix per comunicar informació entre classes.

Interfície: un tipus de classe on només s'especifiquen les signatures dels mètodes però no la seva implementació.

Classe: un fitxer estructurat que conte mètodes i representa lògicament un concepte o funcionalitat.

Bibliografia

[1] Modalitat B CCE

<http://www.fib.upc.edu/fib/empresa/practiques/empresa.html>

[Consulta: 27/02/16]

[2] Everis Home Page

<http://www.everis.com/global/en-US/home/Paginas/home.aspx>

[Consulta: 27/02/16]

[3] MDCS Fujitsu

http://www.fujitsu.com/es/about/resources/news/press-releases/2004/Spain_Fujitsu_Espa_a_Services_desarrollar_una_nueva_aplicac.html

[Consulta: 29/02/16]

[4] JSI Dynasty Technology Group Wincor

http://www.wincor-nixdorf.com/internet/site_EN/EN/Home/homepage_node.html

[Consulta: 02/03/16]

[5] Global JAM Simulator

<http://www.dynasty.es/dynasty/contenidos.item.action?id=8463327&menuId=8463327>

[Consulta: 02/03/16]

[6] Guies de Scrum

<http://www.scrumguides.org/scrum-guide.html>

[Consulta: 28/03/16]

[7] Interface Segmentation Principle o ISP

http://cataleg.upc.edu/record=b1105895~S1*cat

[Consulta: 15/06/16]

[8] Cucumber

<https://cucumber.io/docs>

[Consulta: 16/06/16]

[9] Principi de les set generacions

<http://www.mollylarkin.com/what-is-the-7th-generation-principle-and-why-do-you-need-to-know-about-it-3/>

[Consulta: 19/06/16]

[10] Informe Brundtland

<http://www.un-documents.net/our-common-future.pdf>

[Consulta: 19/06/16]

[11] Apache 2.0 License

<http://www.apache.org/licenses/LICENSE-2.0>

[Consulta: 16/06/16]

[12] GNU General Public License

<https://www.gnu.org/licenses/gpl-3.0.html>

[Consulta: 16/06/16]

[13] MIT License

<https://opensource.org/licenses/MIT>

[Consulta: 16/06/16]

